



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia Eletrônica

Estudo comparativo entre algoritmos de otimização de recursos de satélite para o protocolo DVB-RCS2

Autor: Rodrigo Magalhães Caires
Orientador: Dr. Leonardo Aguayo

Brasília, DF
2020



Rodrigo Magalhães Caires

Estudo comparativo entre algoritmos de otimização de recursos de satélite para o protocolo DVB-RCS2

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Dr. Leonardo Aguayo

Brasília, DF

2020

Rodrigo Magalhães Caires

Estudo comparativo entre algoritmos de otimização de recursos de satélite para o protocolo DVB-RCS2

Monografia submetida ao curso de graduação em (Engenharia Eletrônica) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia Eletrônica).

Trabalho aprovado. Brasília, DF, :

Dr. Leonardo Aguayo
Orientador

Dr. Guillermo Alvarez Bestard
Convidado 1

Dr. Daniel Costa Araújo
Convidado 2

Brasília, DF
2020

Resumo

Estratégias de otimização de recursos são essenciais para a gestão de tráfego em sistemas de telecomunicação, onde a capacidade da rede é limitada e a demanda alta. Somadas a implementação de tecnologias adaptativas que permitem a adequação da rede às necessidades dos usuários, pode-se alcançar altos níveis de qualidade de serviço em termos de flexibilidade e confiabilidade. O atendimento aos usuários pode ser realizado por diferentes critérios que determinam a prioridade de serviço do provedor, sendo os princípios de justiça e eficiência os de características mais atrativas e definidos por objetivos praticamente opostos. Desta forma, estabelece-se como proposta deste trabalho a criação de um algoritmo de alocação de usuários para *superframes* MF-TDMA baseado nas especificações do protocolo DVB-RCS2 e a comparação entre as metodologias igualitária e gananciosa de distribuição de recursos. A ferramenta criada é capaz de separar os acessos em cinco categorias sob diferentes condições de transmissão e otimizar a largura de banda recebida por cada uma. Os usuários são então alocados inicialmente por nível de prioridade, em um segundo momento por ordem de eficiência e finalmente por critérios de justiça almejando uma distribuição proporcionalmente igualitária. Neste trabalho, é introduzido como novidade a utilização e formulação de uma taxa de transferência de recursos dinâmica (Gamma dinâmico) e adaptável com o objetivo de maximizar a velocidade de otimização dos sistemas sem prejudicar a qualidade dos resultados. Os gráficos gerados apresentam o comportamento esperado para ambos estágios, confirmando a reação dual entre os índices e o funcionamento do agendador.

Palavras-chaves: justiça, problema do mochileiro, eficiência, otimização, alocação, MF-TDMA, DVB-RCS2, superframe, adaptação dinâmica.

Abstract

Resource optimization strategies are essential for traffic management in telecommunications systems, where network capacity is limited and demand is high. Added to the implementation of adaptive technologies that allows the adequacy of the network to the users necessities, we are able to reach high levels of quality of service in terms of flexibility and reliability. The treatment of user request can be done by different criteria that determines the priority of service for providers, being the principles of fairness and efficiency the ones with the most desirable characteristics and defined by practically oposite objectives. Therefore, we establish as the main goals in this work the creation of an user allocation algorithm for MF-TDMA superframes based on the specifications of the DVB-RCS2 protocol and the comparison between equitable and greedy resource distribution. The created tool is able to separate the incoming access requests in five categories under different transmission conditions and to optimize the bandwidth received by each category. The users are then allocated initially by level of priority, in a second moment by order of efficiency and finally by a fairness criterion in a proportionally equal fashion. In this work, we present as an innovation the use and formulation of a dynamic and adaptable resource transfer rate (dynamic Gamma) with the objective of maximizing the systems' optimization speeds without harming the quality of the final results. The generated plots show a behaviour just as expected for both phases, confirming the dual reaction between the indexes and the correct operation of the scheduler.

Key-words: fairness, knapsack problem, efficiency, optimization, allocation, MF-TDMA, DVB-RCS2, superframe, dynamic rate adaptation.

Lista de ilustrações

Figura 1 – Modelo do sistema de comunicação DVB-S2/DVB-RCS2. Fonte: (DUPE, 2015)	11
Figura 2 – Diagrama de conexão e alocação de usuários. Fonte: (PIETRABISSA; FIASCHETTI, 2012)	16
Figura 3 – Estrutura geral de um superframe MF-TDMA. Fonte: (LEE et al., 2002)	17
Figura 4 – Constelações utilizadas pelo DVB-RCS2. Fonte: (ETSI TS 101 545-1, 2014)	18
Figura 5 – Exemplo de funcionamento do algoritmo Branch-and-Bound.	28
Figura 6 – Modelo de alocação no superframe MF-TDMA para o DVB-RCS2.	31
Figura 7 – Curvas de otimização de frames para múltiplos valores de Beta.	32
Figura 8 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 1$.	33
Figura 9 – Curvas de otimização de timeslots para $\beta = 2$ e $\gamma = 1$.	33
Figura 10 – Curvas de otimização de timeslots para $\beta = 5$ e $\gamma = 1$.	33
Figura 11 – Curvas de otimização de timeslots para $\beta = 10$ e $\gamma = 1$.	34
Figura 12 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 1$.	35
Figura 13 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 2$.	35
Figura 14 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 5$.	35
Figura 15 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 10$.	36
Figura 16 – Curvas de otimização de timeslots para $\gamma = 2\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.	37
Figura 17 – Curvas de otimização de timeslots para $\gamma = 3\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.	37
Figura 18 – Curvas de otimização de timeslots para $\gamma = 4\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.	37
Figura 19 – Curvas de otimização de timeslots para $\gamma = 5\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.	38
Figura 20 – Curvas de otimização de timeslots para 0% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.	39
Figura 21 – Curvas de otimização de timeslots para 10% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.	39
Figura 22 – Curvas de otimização de timeslots para 20% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.	39
Figura 23 – Curvas de otimização de timeslots para 30% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.	40
Figura 24 – Curvas de otimização de timeslots para sistemas com poucos usuários e alta payload média.	40

Figura 25 – Curvas de otimização de timeslots para sistemas com excesso de usuários e baixa payload média.	41
Figura 26 – Curvas de excesso de usuários vs excesso de payload para cenários de mesma magnitude.	42
Figura 27 – Complexidade temporal relativa ao número de usuários.	43

Lista de abreviaturas e siglas

ACM	- Adaptive Coding and Modulation
BTU	- Bandwidth-Time Unit
CAC	- Connection Admission Control
CMT	- Correction Message Table
DA	- Dedicated Access
DAMA	- Demand Assignment Multiple Access
DRA	- Dynamic Rate Adaptation
DVB-RCS2	- Digital Video Broadcasting - Return Channel via Satellite gen. 2
FMT	- Fade Mitigation Techniques
MF-TDMA	- Multi-Frequency Time-Division Multiple Access
NCC	- Network Control Center
RA	- Random Access
RCST	- Return Channel Satellite Terminal
SNR	- Relação Sinal-Ruído

Lista de símbolos

β	- Parâmetro peso para otimização de canais
γ	- Parâmetro peso para otimização de terminais
ϵ	- Eficiência do terminal t pertencente a DRA c
CH	- Vetor dos canais
CH^*	- Vetor final otimizado dos canais
C	- Quantidade de canais por esquema DRA
c	- ID do esquema DRA
t	- ID de um terminal associado a um esquema DRA
$D_{TS,c}$	- Duração de um timeslot do esquema DRA
D_{SF}	- Duração total do superframe
BL_c	- Burst length do esquema DRA
$L_{c,t}$	- Lucro/Qualidade da conexão do terminal t pertencente a DRA c
GS_c	- Quantidade de símbolos de guarda do esquema DRA
SR_c	- Taxa de símbolos do esquema DRA
$R_{c,t}$	- Requisições em bytes de cada terminal
PL_c	- Tamanho de payload de uma burst do esquema DRA
$N_{TS,c}$	- Números de timeslots disponíveis por canal do esquema DRA
BW_c	- Largura de banda dos canais do esquema DRA
QT_c	- Quantidade de terminais por esquema DRA
$HP_{c,t}$	- Indicador de alta prioridade do terminal t do esquema DRA c
TS_c	- Quantidade de timeslots disponíveis no esquema DRA c
TSR_c	- Quantidade restante de timeslots disponíveis no esquema DRA c
LB	- Lower Bound

UB	- Upper Bound
Z	- Lucro total da otimização
VO	- Vetor reorganizado dos índices de eficiência em ordem decrescente
$X_{c,t}$	- Matriz de alocação dos timeslots
X_c^*	- Vetor de alocação dos timeslots otimizado para o esquema DRA
$X_{c,t}^{req}$	- Matriz de quantidade de timeslots exigidas por terminal
$X_{c,t}^{norm}$	- Matriz normalizada de timeslots por terminal
Y_c^{req}	- Vetor quantidade de timeslots exigidos por esquema DRA
Y_c^{norm}	- Vetor normalizado de timeslots por esquema DRA
$J(X)$	- Índice de otimização dos terminais
$J(Y)$	- Índice de otimização dos canais

Sumário

1	INTRODUÇÃO	11
1.1	Motivação	12
1.2	Objetivos e Métodos	13
1.3	Trabalhos Relacionados	13
2	O PADRÃO DVB-S	15
2.1	Histórico e Características Gerais	15
2.2	Canal de Retorno DVB-RCS2	15
3	ALOCAÇÃO DE RECURSOS EM MF-TDMA	19
3.1	Introdução	19
3.2	Índice de Justiça de Jain	20
3.2.1	Algoritmo com Critério de Justiça	20
3.3	Problema do Mochileiro	23
3.3.1	Algoritmo BKP Greedy	24
3.3.2	Algoritmo MKP Branch-and-Bound	26
4	RESULTADOS DE SIMULAÇÃO	30
4.1	Introdução	30
4.2	Análise dos Parâmetros	31
4.2.1	Beta (β)	32
4.2.2	Gamma (γ)	34
4.2.3	Prioridade de usuários	38
4.2.4	<i>Payload</i> e Quantidade de usuários	40
4.3	Estudo Comparativo dos Algoritmos	42
4.4	Comentários e Conclusões	43
5	CONCLUSÃO E TRABALHOS FUTUROS	45
5.1	Resultados Obtidos	45
5.2	Trabalhos Futuros	45
	REFERÊNCIAS	47

1 Introdução

A popularização dos sistemas de telecomunicação nas últimas décadas, devida especialmente à massificação de tecnologias móveis, tem como consequência uma demanda crescente por recursos e velocidade capazes de suprir um número cada vez maior de usuários. Essas novas exigências forçaram o desenvolvimento de técnicas mais modernas para maximizar a eficiência de utilização dos recursos dos satélites. Entre as inovações, pode-se citar a criação de novos padrões de comunicação dedicados para sistemas dinâmicos e o uso de algoritmos de otimização de rede.

Os sistemas DVB utilizam estratégias híbridas onde os enlaces direto (*forward link*) e reverso (*return link*) são implementados por protocolos de comunicação distintos, diga-se DVB-S2 e DVB-RCS2 respectivamente (ETSI TS 101 545-1, 2014). A separação das funções é justificada pela assimetria do formato DVB, onde na versão S2 o satélite recebe dados de uma única fonte direcionados para vários clientes e para a versão RCS2 as informações partem de múltiplos usuários para uma estação base.

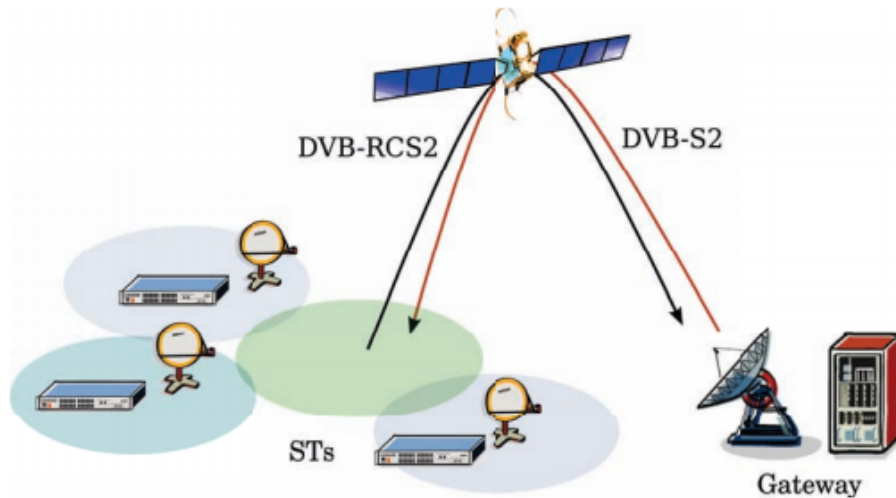


Figura 1 – Modelo do sistema de comunicação DVB-S2/DVB-RCS2. Fonte: (DUPE, 2015)

O protocolo DVB-RCS2 analisado neste trabalho especifica o curso de dados no sentido terminal-satélite e satélite-*gateway*, que compõem o laço de retorno. Este padrão utiliza mecanismos ACM (*Adaptive Coding and Modulation*) e FMT (*Fading Mitigation Techniques*) onde o satélite verifica a qualidade da transmissão de cada terminal e define esquemas de comunicação compatíveis com suas necessidades (PIETRABISSA; FIASCHETTI, 2012). A composição do acesso múltiplo segue o modelo MF-TDMA permitindo a divisão dos recursos em tempo e frequência por meio de *timeslots* (ETSI TS 101 545-1, 2014) em uma estrutura hierárquica de quadros baseada em *superframes*.

A proposta de algoritmos adaptativos prevê mudanças na forma e no fluxo do uso dos recursos da rede, reestruturando o sistema para melhor atender aos usuários considerando suas demandas e restrições. Terminais sob condições meteorológicas ou localizações geográficas diferentes, por exemplo, sofrem de maiores ou menores graus de atenuação de potência útil do sinal, logo operando sob diferentes taxas de relação sinal-ruído (SNR, *signal-to-noise ratio*). Para fornecer o melhor serviço possível, consideram-se diversos tipos de portadoras, cada qual definida por um formato de modulação, taxa de símbolos e taxa de código (*code rate*), agrupando usuários com necessidades semelhantes no que é chamado um esquema DRA (*Dynamic Rate Adaptation*), (PIETRABISSA; FIASCHETTI, 2012). O objetivo destes esquemas DRA é minimizar a taxa de erro das transmissões e ao mesmo tempo maximizar o *throughput* dos usuários considerando a qualidade do sinal e da comunicação e pode assumir modulações de alta eficiência espectral porém mais vulneráveis a interferências como o 16-QAM para usuários com alta SNR ou modulações mais robustas e menos eficientes como o QPSK para baixas SNR.

1.1 Motivação

O controle de tráfego de usuários é uma ferramenta central para todas as áreas da comunicação, onde a capacidade de acesso pode ser limitada em tempo, frequência, canais de transmissão e área de serviço. Com a crescente acessibilidade à meios de comunicação, prover serviço a usuários de maneira rápida e eficiente torna-se não só uma necessidade mas um desafio. A fim de moderar congestionamentos na rede evitando, quando possível, a negação de serviço (DoS - *Denial of Service*) e perdas de dados com conexões de baixa qualidade, visa-se um sistema capaz de otimizar a distribuição de recursos segundo as preferências do provedor da rede.

Com as novas tecnologias 5G e 6G em desenvolvimento, a prospecção torna-se ainda mais promissora com a expectativa de aumento no número de conexões simultâneas (1 milhão por km² para o 5G e até 10 vezes mais para o 6G (CHEN et al., 2020)). Em (LIOLIS et al., 2019) encontram-se alguns cenários possíveis para integração de redes de satélite com sistemas 5G.

Dentro do contexto nacional, a inserção das redes satelitais em conjunto com o cenário 5G/6G viabiliza uma possibilidade interessante para uso de técnicas de escoamento de tráfego devido à vantagem de possibilitar cobertura em áreas remotas das regiões Norte e Nordeste do País. De maneira mais geral, as possibilidades de integração estão evidenciadas pelo grupo de estudo do 3GPP para redes não-terrestres (NTN - *non-terrestrial networks*), em especial nos cenários de *Massive Machine-Type Communications* (mMTC) e *Ultra-Reliable and Ultra-Low-Latency Communications* (URLLC). Em (3GPP TR 38.811 V15.0.0 (2018-06), 2016) e (3GPP TR 22.822 V16.0.0 (2018-06), 2018) definem-se as di-

retrizes gerais para a integração. Para a Release 16 do 3GPP já existem propostas para itens de trabalho relacionados ao tema, a saber: SP 180326 (*Integration of Satellite Access in 5G (5GSAT)*), SP 180505 (*Study on architecture aspects for using satellite access in 5G*) e RP-181370 (*Solutions for NR to support Non Terrestrial Networks*).

1.2 Objetivos e Métodos

O intuito deste trabalho é realizar uma análise comparativa entre duas modalidades de otimização de alocação de recursos de rádio em redes satelitais, uma considerando justiça da distribuição destes recursos entre usuários e a outra considerando lucro ou benefício, sob a perspectiva de alocação de recursos para sistemas MF-TDMA. O sistema em questão tem como base o protocolo DVB-RCS2 responsável pelo enlace reverso da transmissão de vídeo digital e sujeito a adaptações dinâmicas (DRA) capazes de reestruturar os canais de comunicação de acordo com a necessidade dos usuários.

As métricas de comparação entre os algoritmos serão obtidas por intermédio de cálculos executados por um simulador de solicitações de acesso, um algoritmo de otimização de SNR média baseado no problema do mochileiro (*Knapsack Problem* ou KP) (KELLERER; PFERSCHY; PISINGER, 2004) e um algoritmo de otimização de justiça segundo o critério de Jain (JAIN; CHIU; WR, 1984). As análises levarão em conta a variação de parâmetros internos e externos ao sistema e seus respectivos efeitos em ambos os índices. Todo o código foi elaborado em linguagem Python, disponível em <https://github.com/RodrigoMac/OtimizacaoDVB-RCS2>.

1.3 Trabalhos Relacionados

(MARTELLO; TOTH, 1990) apresentam múltiplos algoritmos de uso genérico para cada variante do *Knapsack Problem* por metodologias exatas ou aproximadas. A eficiência e complexidade computacional de cada algoritmo são examinadas teórica e experimentalmente. Em (LEE et al., 2002), encontra-se uma das primeiras tentativas de otimização de *superframe* voltadas para o protocolo DVB-RCS de primeira geração. O algoritmo sugerido aplica penalidades aos usuários de acordo com estimativas de tempo de espera e de perdas de pacotes, sendo o objetivo da função minimizar a penalidade total do *superframe*. Porém, não se relaciona um *tradeoff* custo-benefício característico do *Knapsack Problem*. Em (YAZDANI, 2008), encontra-se uma primeira abordagem para o uso do algoritmo do mochileiro em um sistema MF-TDMA, onde o 'valor' do usuário é uma variável em aberto e pode assumir diversas características do mesmo. O valor é balanceado pelo volume de recursos exigidos pelo usuário, transformando a otimização em uma busca pelas alocações mais eficientes. Entretanto, não há concepção de equidade no compartilhamento de recursos.

Na referência (JUNG; SHIN; RYU, 2019), os autores apresentam um algoritmo de otimização baseado em critérios de justiça para *superframes* DVB-RCS2, porém não há consideração sobre fatores de eficiência. O algoritmo descrito é dividido em duas etapas: a primeira para otimização de *frames* e a segunda para usuários. Em (PIETRABISSA; FIASCHETTI, 2012), analisa-se a obstrução de canal gerada por fenômenos climáticos e como a atenuação resultante na potência dos sinais na região de cobertura do satélite pode provocar mudanças na composição do *superframe* DVB-RCS2. São averiguadas no mesmo trabalho as diferenças entre os desempenhos de estruturas dinâmicas e estáticas do *superframe*.

(KARSU; MORTON, 2015) apresentam uma revisão de trabalhos realizados na área de otimização que concernem tanto eficiência como equidade para diversas classes de problemas, incluindo alocação em telecomunicações. Em um contexto mais próximo, (WANG et al., 2018) formulam um problema de *tradeoff* entre justiça e capacidade para satélites multifeixe em estrutura MF-TDMA, onde a potência de transmissão do satélite é um limitante a ser considerado. A justiça é definida pela igualdade na qualidade das transmissões onde usuários sob maiores níveis de atenuação recebam uma potência maior do satélite visando estabilizar a conexão o que, por sua vez, diminui a potência disponível do satélite que poderia ser usada para atender um número maior de usuários e de forma mais eficiente.

O restante do texto está estruturado da seguinte forma: No Capítulo 2, realiza-se uma breve descrição dos padrões DVB-S e DVB-RCS2. No Capítulo 3, apresentam-se os algoritmos de alocação utilizados para obter os resultados de simulação expostos no Capítulo 4. O Capítulo 5 é dedicado às conclusões e trabalhos futuros.

2 O Padrão DVB-S

2.1 Histórico e Características Gerais

A tecnologia satelital para televisão teve suas primeiras demonstrações durante a década de 1960, sendo precedida pelas redes cabeada. Os primeiros satélites lançados operavam em bandas abaixo de 1 GHz sendo posteriormente substituídas pela banda C (4 a 8 GHz), transmitindo sinais analógicos que eram recebidos por enormes antenas parabólicas. A maior vantagem no desenvolvimento de satélites para TV foi a possibilidade de acesso a regiões remotas onde a conexão cabeada é inacessível ou indisponível.

Introduzido em 1995, o protocolo DVB-S (*Digital Video Broadcast - Satellite*) foi o primeiro a trazer transmissão digital de vídeo via satélite, definindo os padrões para estruturação de *frame*, codificação dos canais e modulação para serviços na faixa de 11/12 GHz. O foco do DVB-S é o fluxo unidirecional de vídeo partindo do provedor para o usuário (*forward link*), ideal para sistemas de televisão pura. A versão DVB-RCS de 1999 foi criada para complementar o DVB-S e permitir a comunicação no sentido usuário-provedor (*return link*), aplicável para TV's com serviços interativos e de internet. O DVB-RCS ainda possui a capacidade de operar individualmente e independentemente do protocolo DVB-S e, como exemplo, pode ser utilizado por centrais de monitoramento para receber transmissão de vídeo ao vivo de diversos terminais via satélite.

Outra versão amplamente utilizada é o DVB-T (*Digital Video Broadcast - Terrestrial*) para estações em solo. Este formato faz o uso de torres de transmissão em frequência de rádio e é comumente empregado por estações locais de notícia e programação regional. Por esse motivo, sinais DVB-T geralmente são multiplexados com sinais de rádio e de internet móvel, compartilhando a mesma banda e as mesmas torres de transmissão. Atualmente, estuda-se a aplicação de sistemas híbridos de redes satelitais e terrestres para serviços de alta velocidade e confiabilidade com foco nas tecnologias 5G e 6G (ZANGAR; HENDAUI, 2018).

2.2 Canal de Retorno DVB-RCS2

Para conectar-se com a rede, um usuário deve inicialmente ser sincronizado e verificar o estado da rede. Após a sincronização, a RCST (terminal RCS) envia uma solicitação de serviço ao NCC (*Network Control Center*), junto de sua demanda de timeslots e da SNR de conexão com a rede, sendo função do módulo CAC (*Connection Admission Control*) garantir ou recusar o acesso. O módulo DAMA (*Demand Assignment Multiple Ac-*

cess) é responsável pelo gerenciamento da capacidade do sistema com base nas demandas dos usuários admitidos pelo CAC. Usuários de alta prioridade exigem que a quantidade de *timeslots* requisitada ao CAC seja integralmente reservada durante toda a conexão, enquanto usuários de baixa prioridade partilham a capacidade remanescente entre si seguindo os critérios de divisão de recursos estabelecidos pelo DAMA. O Dynamic Framing permite uma estruturação dinâmica do bloco MF-TDMA, possibilitando que as portadoras utilizadas pelo mesmo se adaptem às necessidades dos usuários periodicamente. O conjunto entre o CAC e o DAMA forma o *Network Control Center*, abrangendo não só o serviço de alocação de timeslots nos *superframes* como também o envio da CMT (*Correction Message Table*) às RCST's conectadas, fornecendo valores corrigidos de *timing* de sincronização, frequência de *burst* e amplitude de sinal para uma transmissão adequada. O TBTP2 (*Terminal Burst Time Plan gen. 2*) indica aos terminais a frequência, o instante de início e a duração alocadas pelo NCC para que o mesmo possa transmitir.

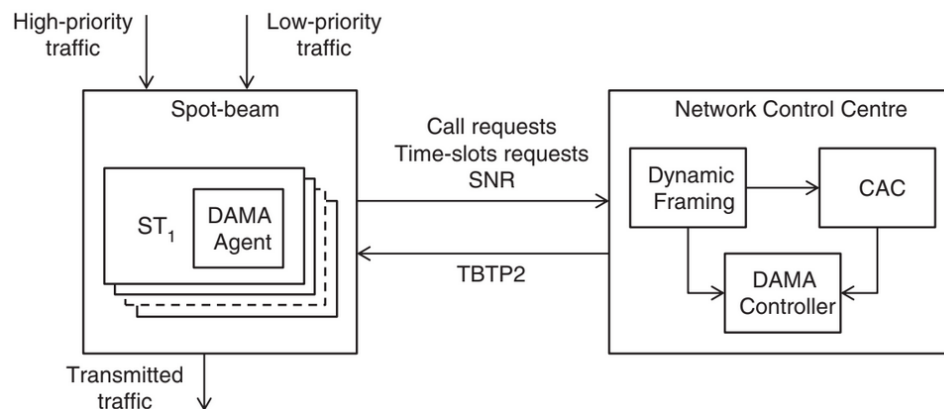


Figura 2 – Diagrama de conexão e alocação de usuários. Fonte: (PIETRABISSA; FIASCHETTI, 2012)

Além dos canais de acesso dedicado (DA), onde *timeslots* são individualmente alocados para cada usuário como mencionado no último parágrafo, existem ainda canais de acesso randômico (RA), com *timeslots* que podem ser acessados simultaneamente por diversos usuários, e canais de acesso misto (DA e RA). Na modalidade RA, como os *timeslots* não são designados para usuários específicos, os mesmos competem entre si para transmitir pacotes, o que resulta frequentemente em colisões destrutivas de sinais quando ambos tentam usar o mesmo *timeslot* ao mesmo tempo. Nesses casos, o *timeslot* é desperdiçado e os usuários recebem um *timeout* aleatório antes de poderem retransmitir o pacote, a fim de evitar novas colisões. Este sistema faz com que cada terminal tenha uma probabilidade de transmitir um sinal durante um instante específico, considerando as chances de que nenhum outro usuário tente transferir pacotes no mesmo instante, ou seja, as chances de uma transmissão bem sucedida. Sendo assim, a eficiência de *throughput* desses canais depende fortemente do método utilizado, podendo atingir um máximo de 37% para o protocolo Slotted-ALOHA ou um valor próximo de 100% para o protocolo

CRDSA (*Contention Resolution Diversity Slotted-ALOHA*), sendo ambos permitidos pelo DVB-RCS2 (MELONI; MURRONI, 2014).

A estrutura do *return link* é organizada em *superframes* que se estendem pelo tempo e pela frequência, sendo esses subdivididos em até 256 *frames* que se distinguem por diferentes tipos, espaços de banda e instantes de começo, podendo variar não só em posição como em tamanho. Os *superframes* são utilizados por motivos de eficiência para o *forward link*. Os *frames*, por sua vez, são compostos por canais e *timeslots* definidos pelo tipo da BTU (*Bandwidth-Time Unit*) associada ao *frame*. O tipo da BTU uniformiza a taxa de símbolos e a largura de banda de todos os canais de um *frame*, mas os *timeslots* podem ser formados por uma quantidade variável de BTU's com um limite de 2048 *timeslots* por *frame*. Os *timeslots* são separados por tempos de guarda que consideram efeitos transientes no sinal e pequenas divergências no *timing* de sincronização, evitando interferências entre *bursts* adjacentes e perdas de pacote. *Superframes* e *frames* tem uma duração máxima de 93,2 segundos.

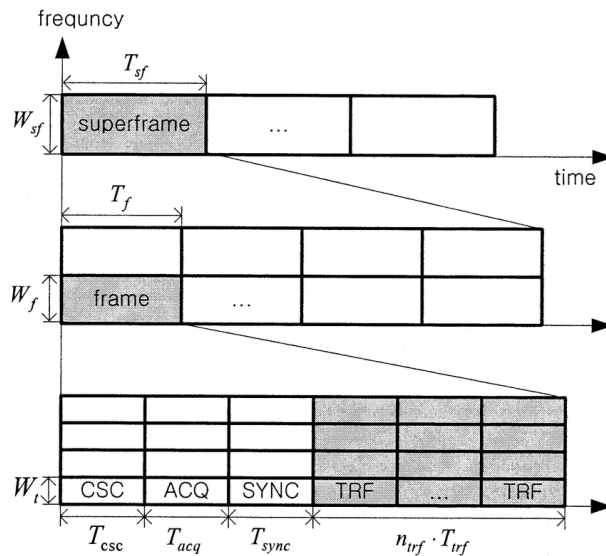
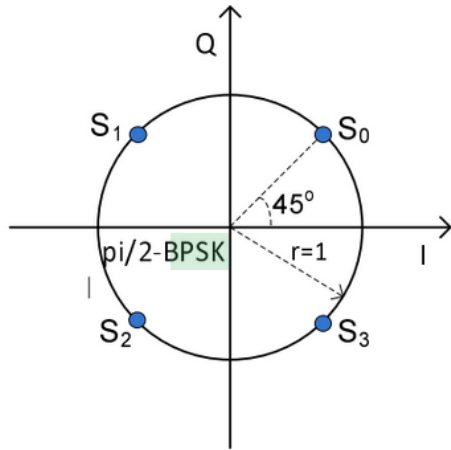
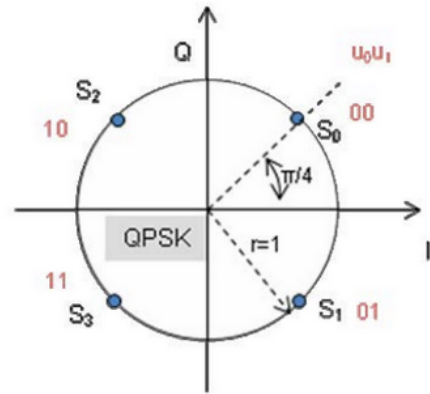


Figura 3 – Estrutura geral de um superframe MF-TDMA. Fonte: (LEE et al., 2002)

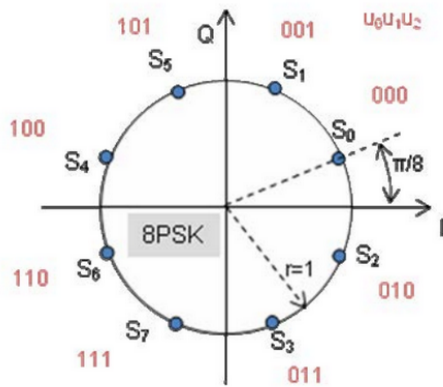
Entre os *bursts* transmitidos pelo usuário na figura 3, cita-se: o CSC (*Common Signaling Channel*) utilizado para informar a versão de software e o identificador do dispositivo; os sinais ACQ (*Acquisition*) e SYNC (*Synchronization*) são ambos utilizados para sincronização do terminal durante e após o *logon*; e o TRF (*Traffic*) contém os pacotes a serem transmitidos para o *gateway*.

São utilizados 4 formatos de modulação pelo protocolo DVB-RCS2, cada um para uma taxa específica de bits por símbolo (bps) e operáveis dentro de um certo intervalo de *code rate*. A modulação $\pi/2$ -BPSK é empregada exclusivamente para cenários de baixa performance, com apenas 1 bps e *code rates* de 1/2 ou 1/3. Como indicado pelo nome, a separação entre os símbolos é de 90° (figura 4) ao invés da constelação mais típica de

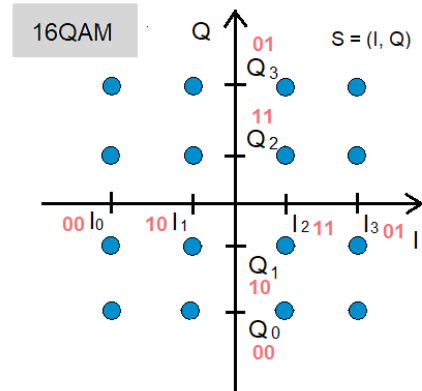
180° . A modulação QPSK é aplicada a uma gama maior de situações sendo o formato com o maior número de configurações (*Waveform ID's*), com 2 bps e *code rates* entre $1/2$ e $7/8$. A modulação 8-PSK é especificada para *code rates* de $2/3$ ou maiores e opera a uma taxa de 3 bps. Por fim, a modulação 16-QAM possui 4 bits por símbolo e é restrita para *code rates* de $3/4$ e maiores. A constelação do 16-QAM é decomposta em dois 4-PAM projetados ortogonalmente (figura 4), sendo o símbolo formado pela concatenação de um ponto do eixo I com um ponto do eixo Q.

(a) $\pi/2$ -BPSK.

(b) QPSK.



(c) 8-PSK.



(d) 16-QAM.

Figura 4 – Constelações utilizadas pelo DVB-RCS2. Fonte: (ETSI TS 101 545-1, 2014)

No capítulo seguinte, as propriedades do *superframe* serão consideradas ao formular a base do problema de otimização estudado. Cada *frame* será representado por um esquema de adaptação dinâmica (DRA) que tem como objetivo minimizar a taxa perda de pacotes de um grupo de usuários sob condições semelhantes de SNR através de mudanças nas taxas de símbolo e código e na largura de banda da portadora. Os *timeslots* de cada *frame* terão durações definidas pela *burstlength* e pela taxa de símbolos estabelecidas pelo esquema DRA, enquanto a quantidade de *timeslots* será limitada pela duração do *superframe*.

3 Alocação de Recursos em MF-TDMA

3.1 Introdução

O estudo de problemas de alocação de recursos é retratado em trabalhos ainda no século 19 e, desde então, já foi adaptado para tratar diversas classes de problemas reais (KELLERER; PFERSCHY; PISINGER, 2004). Um dos casos clássicos trata da otimização de espaço, muito utilizado por transportadoras, onde a capacidade de armazenamento é limitada por volume ou peso e a escolha dos itens deve ser realizada de modo a formar o arranjo mais eficiente possível. Em outro contexto, utilizam-se os mesmos conceitos para organizar uma carteira de ações, onde o investidor tem um orçamento limitado e deve decidir quanto aplicar em cada empresa visando o maior retorno sob o menor risco. Com o amplo desenvolvimento de tecnologias de telecomunicação nas últimas décadas, observa-se o surgimento de uma nova ótica para os problemas de alocação sendo este o objetivo deste trabalho.

A capacidade de transmissão de um satélite é limitada em tempo, frequência e potência, sendo as últimas duas restritas não só por motivos de *hardware* como por regulação. Janelas de tempo são utilizadas por motivos de robustez impedindo que usuários tentem transmitir no mesmo instante causando interferências. Com usuários competindo simultaneamente pelos mesmos recursos, torna-se evidente a necessidade do estabelecimento de regras que definam a distribuição das capacidades do satélite.

Para tanto, pensa-se em maneiras de comparar usuários por meio de parâmetros pertinentes à divisão de recursos. Um critério justo tem como propósito a divisão igualitária dos recursos entre os clientes. Já um critério ganancioso (*greedy*) tem por finalidade atender aos usuários que melhor beneficiem o sistema ao custo de desprezar o restante. Para a primeira situação será utilizado o método de (JAIN; CHIU; WR, 1984) a fim de se estabelecer justiça e para o segundo caso serão testadas duas variantes do problema do mochileiro (KELLERER; PFERSCHY; PISINGER, 2004). Ambos os casos serão analisados nos próximos capítulos para a compreensão de suas qualidades e fraquezas.

Cita-se ainda outras metodologias de otimização não utilizadas neste trabalho como o princípio Rawlsiano (RAWLS, 1971) que tenta maximizar a distribuição de recursos para todos os indivíduos de modo a estabelecer certa equidade. O método, porém, não é capaz de discernir a diferença entre dois casos onde o indivíduo de pior situação receba as mesmas quantidades de recurso mesmo que as distribuições entre os outros indivíduos sejam diferentes. O algoritmo Weighted-Round-Robin (WRR) distribui recursos de maneira cíclica a uma taxa (definida pelo *weight*) específica e pré-estabelecida para

cada receptor e tem como foco a prevenção de *starvation* (negação perpétua de recursos para um ou mais indivíduos) (RENDON-MORALES et al., 2011).

3.2 Índice de Justiça de Jain

O critério de justiça descrito por (JAIN; CHIU; WR, 1984) foi concebido para aplicações de divisão de recursos em engenharia de redes e serve como um indicativo de performance para sistemas orientados a gestão do tráfego de usuários.

Entre os atributos deste método, cita-se: A independência do tamanho da população, devendo ser aplicável para um número finito ou infinito de usuários; A marginalização do índice, sendo um fator de 0 e 1 as representações de um sistema completamente injusto e justo, respectivamente, e; A continuidade, onde variações da alocação de recursos em qualquer escala devem ser perceptíveis ao índice (JAIN; CHIU; WR, 1984).

Como propriedades da função de otimização, os autores definem a necessidade da troca de recursos entre os usuários, seguindo o fluxo de detentores majoritários para detentores minoritários. Propõe-se ainda a delimitação dos níveis de alocação para que nenhum cliente receba menos (mais) recursos que o mínimo (máximo) estipulado, garantindo um certo grau de justiça. Por fim, é apontada a característica côncava da função de otimização onde, após atingir um máximo, nota-se uma tendência de queda no índice de justiça conforme são realizadas mais iterações.

Com base nas características apresentadas, a fórmula proposta para a aferição do nível de justiça e utilizada no desenvolvimento deste algoritmo de otimização segue o modelo exposto abaixo:

$$\text{Maximizar } J(x_f) = \frac{\left(\sum_{t=1}^n \hat{x}_t \right)^2}{n \sum_{t=1}^n \hat{x}_t^2} \quad (3.1)$$

onde $\hat{x}_t = \frac{x_t}{x_t^{req}}$

$J(x_f)$ é o índice de justiça observado pelo grupo de terminais x_f do *frame* f ; \hat{x}_t é a razão 'recursos recebidos' (x_t) por 'recursos solicitados' (x_t^{req}) para o terminal t ; e n é a quantidade de terminais no *frame* f .

3.2.1 Algoritmo com Critério de Justiça

A otimização de *frames* procura os esquemas DRA com a maior e a menor proporção de quantidade de 'canais recebidos' por 'canais necessários'. Em seguida determina-se a diferença proporcional de largura de banda (∇BW) entre os esquemas encontrados. O parâmetro Beta (β) indica quantas proporções de banda serão transferidas entre os *frames*

por iteração. O *frame* com maior proporção de recursos concede então uma quantidade de banda igual a $\beta \cdot \nabla BW$ para o *frame* de menor proporção. Por fim, o índice de justiça entre *frames* é recalculado para a nova distribuição de recursos e os processos se repetem por um número definido de iterações.

Um procedimento semelhante é utilizado para a otimização dos *timeslots*, onde os usuários são buscados pela maior e menor razão '*timeslots* recebidos' por '*timeslots* solicitados'. Não há diferença entre os tamanhos dos *timeslots* de um mesmo *frame* e a taxa de transferência é unicamente definida pelo parâmetro Gamma (γ). O usuário com a maior quantidade proporcional de recursos doa γ *timeslots* para o usuário menos favorecido do sistema. Recalcula-se então o índice de justiça entre os usuários e reiniciam-se os procedimentos até atingir o resultado desejado ou o máximo de iterações.

A inicialização dos canais deve respeitar o tamanho do *superframe*, logo a soma das larguras de banda concedidas para cada *frame* deve ser menor ou igual à largura de banda total disponível. Já a alocação inicial de usuários deve ser limitada pela quantidade de *timeslots* disponíveis por *frame*.

O algoritmo 1 sugerido é baseado no trabalho desenvolvido por (JUNG; SHIN; RYU, 2019). Como diferencial, cita-se a implementação do parâmetro Gamma dinâmico e a inclusão de usuários prioritários. A adição do parâmetro Gamma dinâmico resultará em uma taxa de transferência de recursos adaptável à quantidade de usuários e de *timeslots* disponíveis para cada esquema DRA em qualquer momento e será analisado mais a fundo na seção 4.2.2. Diferentes níveis de prioridade afetam a relação entre usuários e, consequentemente, o resultado final do algoritmo de justiça e seus efeitos serão analisados na seção 4.2.3.

No algoritmo 1, a função `where()` retorna as posições no vetor que contenham o valor pesquisado; As funções `max()` e `min()` retornam os valores máximos e mínimos de um vetor, respectivamente; A função `lcm()` retorna o mínimo múltiplo comum de 2 valores; `linspace()` gera um vetor de números linearmente espaçados; `plot()` desenha a curva formada pelo conjunto de pontos (x,y); O operador `./` realiza a divisão entre elementos de mesma posição de duas matrizes distintas.

Algoritmo 1 Algoritmo de otimização de justiçaInputs do algoritmo: C , D_{SF} , SR_c , BL_c , GS_c , $R_{c,t}$, QT_c , PL_c , BW_c , $HP_{c,t}$, β e γ .

```

1: Inicializar vetor  $CH$  para todo  $c$  e  $J_{Ymax} = 0$ 
2:  $D_{TS,c} = (BL_c + GS_c) ./ SR_c$ 
3:  $N_{TS,c} = D_{SF} ./ D_{TS,c}$ 
4:  $X_{c,t}^{req} = R_{c,t} ./ PL_c$ 
5:  $Y_c^{req} = \sum_{t=0}^{QT_c} X_{c,t}^{req}$ 
6:  $Y_c^{norm} = CH_c N_{TS,c} ./ Y_c^{req}$ 
7: Otimização dos frames:
8: for  $j$  in  $range(i_{Ymax})$  do
9:    $c_{max} = where(max(Y_c^{norm}))$  e  $c_{min} = where(min(Y_c^{norm}))$ 
10:   $\nabla_{CHmax} = lcm(BW[c_{max}], BW[c_{min}]) / BW[c_{max}]$ 
11:   $\nabla_{CHmin} = lcm(BW[c_{max}], BW[c_{min}]) / BW[c_{min}]$ 
12:   $CHc_{max} = CHc_{max} - \beta \nabla_{CHmax}$  e  $CHc_{min} = CHc_{min} + \beta \nabla_{CHmin}$ 
13:   $Y_c^{norm} = CH_c * N_{TS,c} ./ Y_c^{req}$ 
14:   $J(Y) = (\sum_{c=0}^C Y_c^{norm})^2 / C \sum_{c=0}^C (Y_c^{norm})^2$ 
15:   $J(Y)_{mem}[j] = J(Y)$ 
16:  if  $J(Y) > J_{Ymax}$  then
17:     $J_{Ymax} = J(Y)$ 
18:     $CH^* = CH$ 
19:  end if
20: end for
21: Alocação de usuários prioritários:
22: for  $k$  in  $range(C)$  do
23:   for  $j$  in  $range(QT_c)$  do
24:    if  $(HP[k, j] == 1 \text{ AND } X_{c,t}^{req}[k, j] \leq TSR_c)$  then
25:       $X_{c,t}[k, j] = X_{c,t}^{req}[k, j]$ 
26:       $TSR_c = X_{c,t}^{req}[k, j]$ 
27:    end if
28:   end for
29: end for
30: Otimização dos timeslots:
31: for  $k$  in  $range(C)$  do
32:   Inicializar vetor  $X_c$  para todo  $t$  e  $J_{Xmax} = 0$ 
33:    $X_{c,t}^{norm} = X_{c,t} ./ X_{c,t}^{req}$ 
34:   for  $j$  in  $range(i_{Xmax})$  do
35:     $t_{max} = where(max(X_{c,t}^{norm}))$  e  $t_{min} = where(min(X_{c,t}^{norm}))$ 
36:     $X_{c,t_{max}} = X_{c,t_{max}} - \gamma$  e  $X_{c,t_{min}} = X_{c,t_{min}} + \gamma$ 
37:     $X_{c,t}^{norm} = X_{c,t} ./ X_{c,t}^{req}$ 
38:     $J(X) = (\sum_{t=0}^{QT} X_{c,t}^{norm})^2 / QT \sum_{t=0}^{QT} (X_{c,t}^{norm})^2$ 
39:     $J(X)_{mem}[j, k] = J_{Xmax}$ 
40:    if  $J(X) > J_{Xmax}$  then
41:       $J_{Xmax} = J(X)$ 
42:       $X_c^* = X_c$ 
43:    end if
44:   end for
45: end for
46:  $ty = linspace(0, i_{Ymax} - 1, 1)$ 
47:  $tx = linspace(0, i_{Xmax} - 1, 1)$ 
48:  $plot(ty, J(Y)_{mem})$ 
49:  $plot(tx, J(X)_{mem})$ 

```

3.3 Problema do Mochileiro

O *Knapsack Problem* (KP) é um problema de otimização de recursos que considera três parâmetros: o peso ou custo de um item, o benefício ou lucro ao possuir o item, e a capacidade ou orçamento disponível para a obtenção de itens (KELLERER; PFERSCHY; PISINGER, 2004). A solução do problema é dada pela escolha de itens que maximize o lucro obtido sem que a soma dos pesos exceda a capacidade do sistema, de acordo com a equação 3.2. Matematicamente, pode ser definido da seguinte forma: é dada uma mochila com capacidade c e é dado um conjunto de itens $\mathbb{N} := \{1, \dots, n\}$, cada item com valor p_j e peso w_j . O problema é encontrar o subconjunto de \mathbb{N} tal que o valor total dos itens seja maximizado, sem exceder a capacidade c . Para essa versão do problema do mochileiro, a variável x_j pode assumir apenas os valores 0 e 1 representando a recusa ou o aceite de um item respectivamente.

$$\begin{aligned} \text{Maximizar } P(x) &= \sum_{j=1}^n p_j x_j \\ \text{sujeito a } \sum_{j=1}^n w_j x_j &\leq c \\ x_j &\in \{0, 1\}, j = 1, \dots, n. \end{aligned} \quad (3.2)$$

A otimização de algoritmos do mochileiro é facilitada pela utilização do parâmetro eficiência, definido pela quantidade de lucro gerada pelo item por cada unidade de peso e representada pela equação 3.3. A organização dos usuários por ordem decrescente de eficiência possibilita que o algoritmo teste as melhores opções em primeiro momento e pode evitar que soluções ruins sejam testadas caso as estimativas de lucro destas sejam menores que o já obtido em soluções anteriores, possivelmente diminuindo o tempo de execução do algoritmo.

$$\epsilon_j = \frac{p_j}{w_j} \quad (3.3)$$

Para adaptar o KP ao problema de alocação de usuários no *superframe* é necessário, primeiramente, definir os parâmetros de entrada da otimização. Considerando o *superframe* como a mochila e os usuários como os itens a serem alocados, temos que: A capacidade pode ser entendida como a quantidade de *timeslots* disponíveis em um dado canal ou esquema DRA, sendo esses compartimentos da mochila; Dessa forma, define-se que o peso de um item é a quantidade de *timeslots* pedida pelo usuário; Por fim, o lucro deve refletir uma característica desejável e quantizável do usuário sendo, neste caso, associada à Relação Sinal-Ruído (SNR). No entanto, utilizaremos a SNR como eficiência pois a mesma pode ser pensada como um benefício proporcional para cada unidade de peso, ou seja, quanto mais *timeslots* forem concedidos para um usuário com alta SNR, maior

será a qualidade média de comunicação observada pelo satélite e então maior deverá ser o lucro observado.

Tabela 1 – Comparativos entre o problema do mochileiro e a estrutura MF-TDMA.

Problema do Mochileiro	Mochila	Compartimento da mochila	Item	Eficiência	Peso
MF-TDMA	Superframe	Esquema DRA	Usuário	SNR	Timeslots pedidos

Dentre as diversas variantes do problema do mochileiro, testou-se as duas soluções de maior afinidade ao problema de alocação de usuários. Algoritmos *Bounded Knapsack* são recomendados para situações com uma única mochila e itens que podem possuir um número finito de cópias (mesmo lucro e peso). Algoritmos *Multiple Knapsack* consideram várias mochilas nas quais os itens podem ser distribuídos de modo a otimizar o lucro e a capacidade de cada uma.

Outras versões do problema foram descartadas após análises mais profundas por não se adequarem bem às expectativas. A variante 2-KP trata de uma limitação bidimensional na capacidade do sistema muito semelhante ao observado na estrutura MF-TDMA. No entanto, não há vantagem na utilização deste método pois usuários não podem ser alocados para frequências distintas durante o mesmo instante de tempo, o que resultaria em uma restrição adicional ao problema e em um aumento na dificuldade desta já complexa solução.

O *Multiobjective Knapsack Problem* (MOKP), por outro lado, considera dois objetivos a serem otimizados ao invés de um. O intuito seria estabelecer uma meta para eficiência e outra para equidade porém logo percebemos que ambos os parâmetros lucro e custo para justiça seriam diretamente proporcionais à quantidade de *timeslots* alocadas para um usuário. O recebimento de *timeslots* deveria representar um aumento na justiça, neste caso diretamente relacionada ao lucro, ao mesmo passo que aumentaria o peso, fazendo o índice constante para qualquer quantidade de recursos concedidos e tornando o sistema inotimizável ou exigindo uma definição não-linear de lucro, novamente aumentando a complexidade da solução.

3.3.1 Algoritmo BKP Greedy

O primeiro algoritmo é a solução de um *Bounded Knapsack Problem* (BKP) executado pelo método *Greedy*. Para melhor entendimento, o uso do termo 'canal' neste documento será referente a uma largura de banda onde os usuários compartilham a mesma frequência central em diferentes instantes de tempo. A variante BKP facilita a otimização em situações com vários itens repetidos e, neste caso, considerou-se cada *timeslot* pedido por um usuário como a cópia de um item. Por simplificação, considera-se um único canal

com capacidade igual à soma das capacidades de todos os canais de um esquema DRA. Os usuários são então ordenados por eficiência (SNR) decrescente e alocados de maneira gananciosa até a quantidade de *timeslots* alocados se igualar à capacidade do canal. Se a capacidade remanescente do canal for menor que a quantidade de recursos pedidos pelo usuário não alocado de maior eficiência, o mesmo é alocado parcialmente recebendo todos os *timeslots* restantes do canal. Este algoritmo retorna sempre o maior nível de SNR média possível de um sistema e a solução final permite que os usuários sejam alocados para até 2 canais distintos, o que é permitido pelo protocolo DVB-RCS2.

A equação 3.4 descreve o problema de otimização para o BKP. Devido a cada *timeslot* ser considerado como a cópia de um item e como todos os *timeslots* de um esquema DRA possuem o mesmo tamanho, pode-se simplificar a variável w_j da equação abaixo como igual a 1 e o 'peso' dos usuários é então definido pela quantidade x_j de *timeslots* recebidos que pode assumir valores entre 0 e b_j , onde b_j representa a quantidade total de *timeslots* pedidos (ou cópias do item).

$$\begin{aligned} \text{Maximizar } P(x) &= \sum_{j=1}^n p_j x_j \\ \text{sujeito a } \sum_{j=1}^n w_j x_j &\leq c \\ 0 \leq x_j &\leq b_j, x_j \text{ inteiro}, j = 1, \dots, n. \end{aligned} \tag{3.4}$$

No algoritmo 2, a variável TS representa a capacidade total de *timeslots* de um esquema DRA enquanto TSR é a quantidade restante de recursos não alocados; ϵ é a eficiência e representa a SNR dos usuários; `argsort()` é uma função que retorna a sequência de índices capazes de reorganizar um vetor em ordem decrescente; VO é uma variável utilizada para reorganizar os outros vetores do algoritmo em ordem decrescente de eficiência; e Z é a soma das SNR's dos usuários alocados multiplicado pela quantidade de *timeslots* concedidos para os mesmos e representa a função de maximização P(x). O algoritmo proposto é de desenvolvimento próprio e baseado na descrição do *Bounded Knapsack Problem* de (KELLERER; PFERSCHY; PISINGER, 2004).

Algoritmo 2 Algoritmo de otimização de SNR média BKP GreedyInputs do algoritmo: $\epsilon_{c,t}$ e TS_c .

```

1: for  $c$  in  $\text{range}(C)$  do
2:   Inicializar  $TSR_c[c] = TS_c[c]$ 
3:   Obter vetor de índices para  $\epsilon$  organizado em ordem decrescente
4:    $VO_{c,t} = \text{argsort}(\epsilon_{c,t})$ 
5:   Organizar  $X_{c,t}^{req}$  e  $X_c^*$  em ordem decrescente de  $\epsilon$ 
6:   for  $t$  in  $\text{range}(QT_c)$  do
7:      $\epsilon_{c,t}[c, t] = \epsilon_{c,t}[c, VO[t]]$ 
8:      $X_{c,t}^{req}[c, t] = X_{c,t}^{req}[c, VO[t]]$ 
9:      $X_c^*[c, t] = X_c^*[c, VO[t]]$ 
10:  end for
11:  for  $t$  in  $\text{range}(QT_c)$  do
12:    if  $X_{c,t}^{req}[c, t] \leq TSR_c$  then
13:       $X_c^*[c, t] = X_{c,t}^{req}[c, t]$ 
14:       $Z+ = \epsilon_{c,t}[c, t] * X_c^*[c, t]$ 
15:       $TSR_c- = X_{c,t}^{req}[c, t]$ 
16:    else
17:       $X_c^*[c, t] = TSR_c$ 
18:       $Z+ = \epsilon_{c,t}[c, t] * TSR_c$ 
19:       $TSR_c = 0$ 
20:    end if
21:  end for
22: end for

```

3.3.2 Algoritmo MKP Branch-and-Bound

O segundo algoritmo é uma variante *Multiple Knapsack Problem* (MKP) solucionado pela metodologia *Branch-and-Bound* e é uma adaptação baseada no algoritmo desenvolvido por (HOROWITZ; SAHNI, 1984). O algoritmo MKP permite a solução de múltiplos sistemas separadamente. Sendo assim, cada canal de um esquema DRA é considerado como um sistema próprio e cada um é resolvido individualmente. Como o MKP é uma extensão do problema genérico do mochileiro, os usuários devem ser ou completamente alocados ou recusados pelo sistema, ou seja, uma concessão parcial dos recursos pedidos não é considerada. Por esse motivo, o algoritmo *Greedy* não é capaz de retornar uma solução ótima para o MKP. O mecanismo *Branch-and-Bound* faz uma busca em árvore seguindo o procedimento *Depth-First* e calcula limites superiores e inferiores (*Bounds*) capazes de prever e eliminar ramos que não resultarão em uma solução ótima. Quando a solução para o primeiro canal é encontrada, os usuários alocados para o mesmo são removidos da otimização e repetem-se os processos até que todos os canais sejam otimizados. A solução final do algoritmo MKP retorna menores níveis de SNR média quando comparado ao BKP, mas cada usuário é exclusivamente alocado para um único canal. A equação 3.5 descreve as particularidades do MKP como uma quantidade m de mochilas

de capacidades c_i e a alocação do item x_{ij} na mochila i .

$$\begin{aligned}
 &\text{Maximizar } P(x) = \sum_{i=1}^m \sum_{j=1}^n p_j x_{ij} \\
 &\text{sujeito a } \sum_{j=1}^n w_j x_{ij} \leq c_i, i = 1, \dots, m. \\
 &\quad \sum_{i=1}^m x_{ij} \leq 1, j = 1, \dots, n. \\
 &\quad x_j \in \{0, 1\}, i = 1, \dots, m, j = 1, \dots, n.
 \end{aligned} \tag{3.5}$$

Algoritmo 3 Algoritmo de otimização de SNR média MKP Branch-and-Bound

Inputs do algoritmo: $L_{c,t}$ e TS_c .

```

1: for  $c$  in  $\text{range}(C)$  do
2:   Inicializar  $TSR_c[c] = TS_c[c]$ ,  $X_{c,t} = 0$ ,  $X_c^* = 0$ ,  $Z = 0$ ,  $LB = -\inf$ ,  $UB = 0$ ,  $t = 0$  e  $pos = 0$ 
3:   while  $True$  do
4:     Forward Step
5:     for  $t$  in  $\text{range}(pos, QT_c)$  do
6:       if  $X_{c,t}^{req}[c, t] \leq TSR_c$  then
7:          $X_{c,t}[c, t] = 1$ 
8:          $Z+ = L_{c,t}[c, t]$ 
9:          $TSR_c[c]- = X_{c,t}^{req}[c, t]$ 
10:      end if
11:      Upperbound
12:       $UB = Z + (TSR_c * L_{c,t}[c, t] / X_{c,t}^{req}[c, t])$ 
13:    end for
14:    Update da solução
15:    if  $UB > LB$  then
16:       $LB = Z$ 
17:       $X_c^* = X_{c,t}$ 
18:       $TSR_c[c] = X_{c,t}^{req}[c, t]$ 
19:    end if
20:    Backtracking
21:     $pos = \text{where}(X_{c,t} == 1)$ 
22:    if  $\text{size}(pos) == 0$  then
23:      break
24:    else
25:       $pos = \max(pos)$ 
26:       $X_{c,t}[c, pos] = 0$ 
27:       $TSR_c[c]+ = X_{c,t}^{req}[c, pos]$ 
28:       $Z- = L_{c,t}[c, pos]$ 
29:       $pos+ = 1$ 
30:    end if
31:  end while
32: end for

```

No algoritmo 3, a função **Forward Step** tenta alocar um usuário na capacidade restante do canal. Se a mesma for insuficiente, a função recusa completamente o pedido deste usuário e testa a capacidade para o próximo na fila; A variável **Upperbound** (UB) atualiza a meta de otimização que o algoritmo tentará atingir. A variável **Lowerbound** (LB) indica o melhor nível de otimização alcançado até o presente; A função **Backtracking** tenta remover o último usuário alocado pela solução e avançar a posição para o próximo da fila para que a função **Forward Step** possa analisar outro ramo da árvore. Se não for possível, isso significa que não há mais ramos para testar e a otimização é encerrada.

A tabela 2 descreve um exemplo do problema do mochileiro para uma mochila de capacidade $c = 9$ e 7 itens j organizados em ordem decrescente de eficiência. A figura 5 demonstra a busca realizada pelo algoritmo *Branch-and-Bound* no exemplo citado. As setas em vermelho representam **Forward Steps** enquanto as setas em azul representam **Backtracking** que ocorrem sempre ao final de um ramo, seja por não existirem mais possibilidades ou pelas possibilidades existentes não serem capazes de retornar uma solução ótima. Valores sobrelinhados dentro dos nós representam o **Upperbound** e oferecem uma estimativa da melhor solução alcançável ao se aprofundar em seus ramos, enquanto os valores sublinhados representam a **Lowerbound** e armazenam a melhor solução obtida pelo algoritmo até a iteração atual. Quando o valor do **Upperbound** de um nó é menor que o valor do **Lowerbound** isso significa que o nó não tem possibilidades de oferecer uma solução melhor que a já existente e não há motivos para testá-lo mais a fundo.

Tabela 2 – Exemplo de problema do mochileiro.

$c = 9$							
j	1	2	3	4	5	6	7
p_j	6	5	8	9	6	7	3
w_j	2	3	6	7	5	9	4

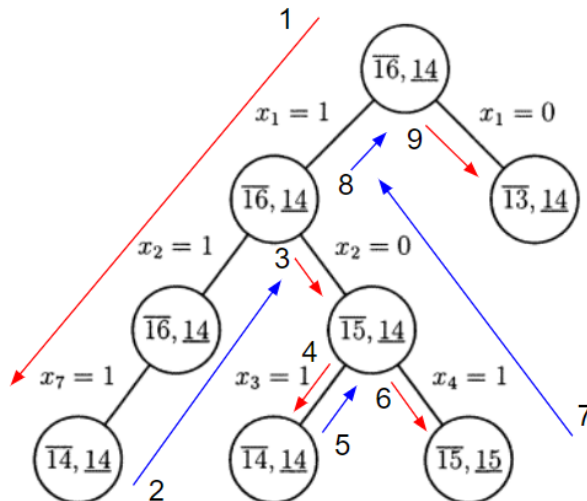


Figura 5 – Exemplo de funcionamento do algoritmo Branch-and-Bound.

Como comparação, o resultado obtido pelo método *Greedy* sob as restrições do algoritmo MKP mencionadas anteriormente seria exatamente igual a seta vermelha de índice 1 da figura 5, ou seja, o subconjunto de itens $\{1, 2, 7\}$ atingindo uma solução com 14 de lucro. Por outro lado, o método *Branch-and-Bound* encontra que a melhor solução é dada pelo subconjunto $\{1, 4\}$, obtendo 15 de lucro. Sob essas circunstâncias estima-se que, no pior caso, o algoritmo *Greedy* é capaz de retornar uma solução com apenas 50% do valor ótimo (KELLERER; PFERSCHY; PISINGER, 2004).

Mesmo com o uso do método *Branch-and-Bound* aumentando consideravelmente a qualidade da solução do algoritmo MKP, o mesmo só é capaz de retornar resultados piores ou iguais ao algoritmo BKP e exige um tempo de execução maior. Seguindo o exemplo da tabela 2, o algoritmo BKP alocaria 2 *timeslots* para o usuário 1, 3 *timeslots* para o usuário 2 e 4 *timeslots* para o usuário 3 totalizando a capacidade do sistema e atingindo uma solução igual a 16,33. Para este exemplo, o BKP retornaria a solução em 3 iterações enquanto o algoritmo MKP precisa de 8 iterações como visto na figura 5. Por esses motivos, decidiu-se por usar o algoritmo BKP em conjunto com o algoritmo Fairness para realizar a divisão de recursos do simulador.

4 Resultados de Simulação

4.1 Introdução

Optou-se pela linguagem Python para a construção do simulador, sendo uma alternativa gratuita com capacidade gráfica e ampla base documentada de bibliotecas e pacotes. O intuito da simulação é criar um ambiente de testes dinâmico capaz de imitar os inputs de um sistema real, expondo os algoritmos à diversas situações coerentes e averiguando seus desempenhos.

Os esquemas DRA simulados são definidos segundo a tabela 3. A duração dos *timeslots* de cada esquema é diretamente proporcional à burst length e inversamente proporcional à symbol rate. Logo, a largura de banda é dobrada enquanto a quantidade de *timeslots* por largura de banda é diminuída em 25% ao se passar de um esquema inferior para o esquema diretamente superior. Desta forma a razão 'quantidade de *timeslots*' por 'largura de banda' é diferente para cada esquema.

Tabela 3 – Propriedades dos esquemas DRA simulados.

Esq. DRA	Largura do Canal (kHz)	Symbol Rate (símb/ms)	Waveform ID	Burst Length (símbolos)	Modulação	Code Rate
1	78	128	13	1616	QPSK	1/3
2	156	192	14	1616	QPSK	1/2
3	312	288	15	1616	QPSK	2/3
4	624	432	16	1616	QPSK	3/4
5	1248	648	17	1616	QPSK	5/6

O cenário padrão de testes será definido pelos parâmetros $\beta = 1$, $\gamma = 1$, e 0% de chance de usuários prioritários, exceto quando informado o contrário. A quantidade de usuários, suas respectivas *payloads* e SNR's são randomizadas a cada rodada da simulação. Utilizou-se uma distribuição de Poisson com $\lambda = 30$ para simular a quantidade de usuários por esquema DRA e uma distribuição uniforme entre 0 e 70,000 para simular a *payload* em bytes solicitada por cada usuário. A SNR dos terminais é representada em níveis entre 0 e 100 definidos aleatoriamente onde valores próximos a 0 indicam que o usuário possui o mínimo de SNR permitida pelo esquema DRA atual e está prestes a ser transferido para um formato de transmissão mais robusto e de menor eficiência. Analogamente, valores próximos a 100 indicam que o mesmo se encontra no limite superior de SNR do esquema DRA atual e pode ser transferido para um formato de maior eficiência. A duração do *superframe* é fixada em 1 segundo e a quantidade de símbolos de guarda é igual a 4 para todos os esquemas DRA.

Os gráficos $J(x)$ e $P(x)$ apresentados nesse capítulo são iniciados após o término do algoritmo do mochileiro e cada ponto representa os efeitos de uma iteração do algoritmo de justiça nos respectivos índices. Sendo assim, a iteração de número 0 nos gráficos representa os valores de justiça e SNR média obtidos ao final do algoritmo do mochileiro. Para fins de comparação, os valores de SNR média são normalizados para valores entre 0 e 1, onde 1 representa o maior valor obtido pelo algoritmo. Dessa forma, será observada uma queda percentual no nível de SNR média dos esquemas DRA durante a execução do algoritmo de justiça.

Na figura 6, é apresentado um modelo de alocação para 12 usuários em diferentes formatos de transmissão no *superframe*. Ressalta-se que um *frame* é formado pelo conjunto de canais de um esquema DRA, ou seja, na figura 6 os canais 1 e 2 da DRA 1 são parte de um *frame*. Menciona-se também que um usuário pode ser alocado para dois canais distintos de um frame em instantes diferentes de tempo, como visto com U3.

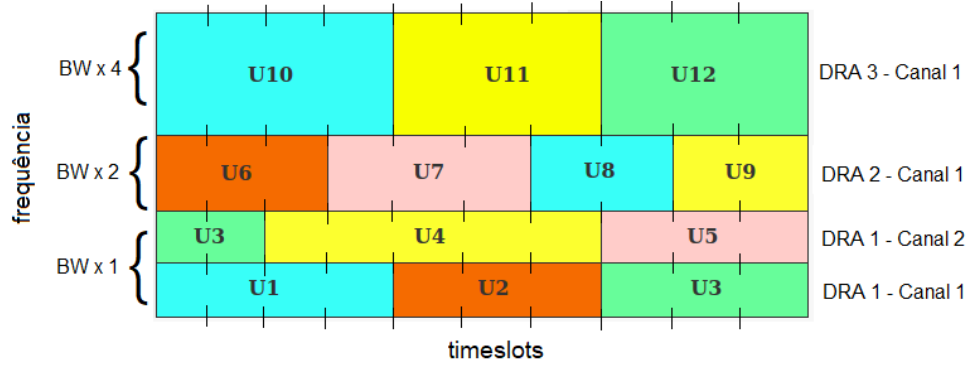


Figura 6 – Modelo de alocação no superframe MF-TDMA para o DVB-RCS2.

4.2 Análise dos Parâmetros

Algumas variáveis do algoritmo apresentam impacto significativo na forma final das curvas de justiça e SNR média para canais e usuários, afetando a velocidade e valor de convergência e a dispersão entre os esquemas DRA. A seguir, serão analisados 5 desses parâmetros e suas influências.

Por motivos de clareza, pode-se pensar no índice de lucro $P(x)$ como a qualidade média de conexão dos usuários com a rede. Valores próximos de 0% indicam que os usuários operam no limite inferior de SNR do esquema DRA e estão prestes a serem transferidos para um esquema inferior e de menor eficiência. De maneira semelhante, índices próximos de 100% sugerem que os usuários se encontram no limite superior de SNR do esquema DRA e podem ser realocados para esquemas DRA mais altos e de maior eficiência.

4.2.1 Beta (β)

O parâmetro Beta regula a quantidade de canais transferida entre dois esquemas DRA por iteração. Valores baixos de Beta realizam um ajuste mais fino e mais lento enquanto valores mais altos convergem mais rápido ao preço de resultados piores. Apesar de ser um fator relacionado à otimização de canais, esse acaba afetando indiretamente a otimização de usuários já que a quantidade de *timeslots* em um esquema depende da quantidade de banda disponível.

Para a análise da otimização de canais segundo o critério de justiça, inicializou-se toda a banda de frequência disponível para um único esquema DRA, representando o caso mais injusto possível de distribuição de recursos. A distribuição inicial é de (256, 0, 0, 0, 0) canais para os esquemas DRA de largura de banda (78, 156, 312, 624, 1248) KHz, totalizando 20 MHz de banda disponível. A influência do parâmetro Beta varia para diferentes proporções de banda disponível e largura de canais. Os gráficos da figura 7 demonstram o impacto de Beta na solução ao longo das iterações do algoritmo.

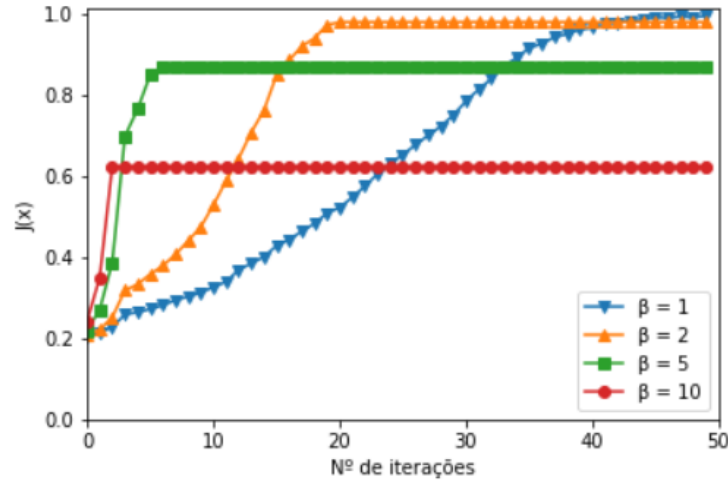
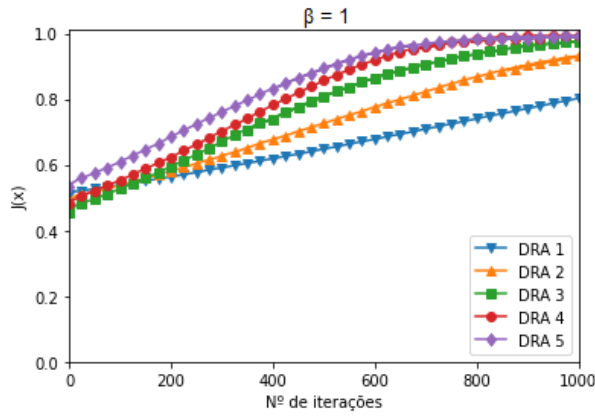


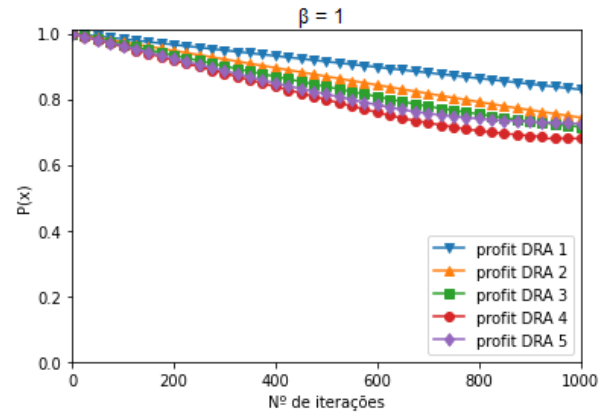
Figura 7 – Curvas de otimização de frames para múltiplos valores de Beta.

Embora o melhor resultado seja obtido através do ajuste mais lento, pode-se reduzir significativamente a quantidade de iterações necessárias caso a distribuição inicial de canais parta de um ponto mais homogêneo, diga-se para casos com larguras de banda iniciais próximas de 4 MHz para cada esquema DRA. Deste modo, consegue-se flexibilizar o *trade-off* velocidade-desempenho alcançando uma excelente otimização para um baixo número de iterações.

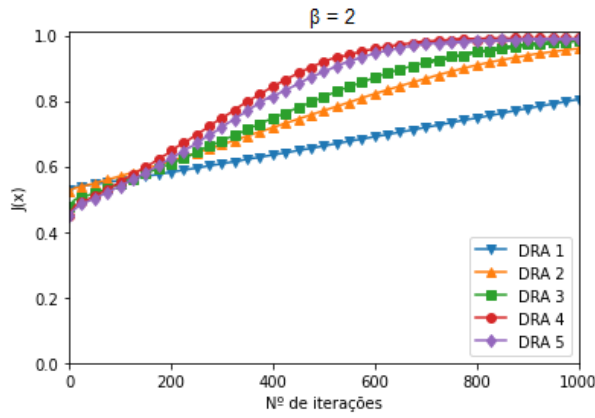
Beta também afeta a divergência inicial de justiça e a divergência final de SNR média entre esquemas DRA durante a etapa de otimização de usuários. Este efeito ocorre pois Beta aumenta a quantidade mínima de banda transferida entre esquemas DRA que, por consequência, afeta a quantidade final de *timeslots* de cada um gerando *overshoots* e *undershoots* que inviabilizam resultados justos.



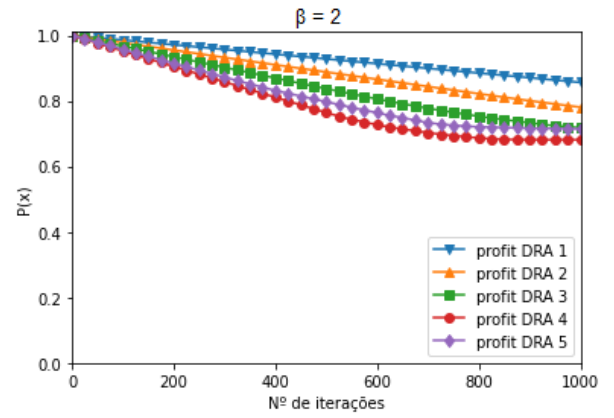
(a) Justiça.



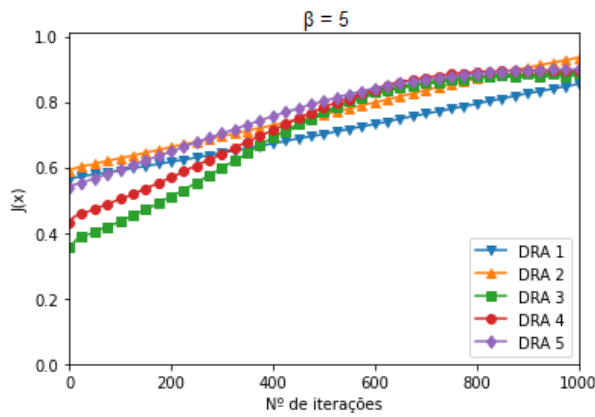
(b) Eficiência.

Figura 8 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 1$.

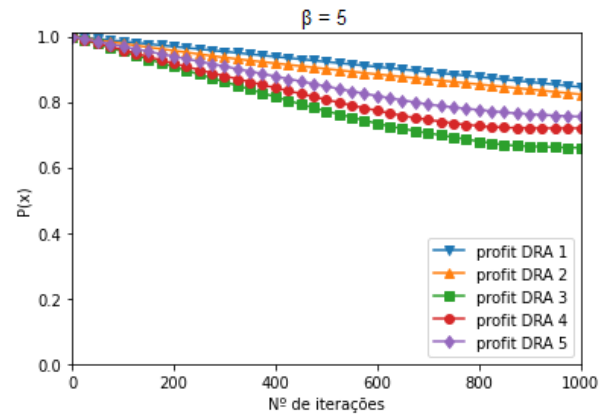
(a) Justiça.



(b) Eficiência.

Figura 9 – Curvas de otimização de timeslots para $\beta = 2$ e $\gamma = 1$.

(a) Justiça.



(b) Eficiência.

Figura 10 – Curvas de otimização de timeslots para $\beta = 5$ e $\gamma = 1$.

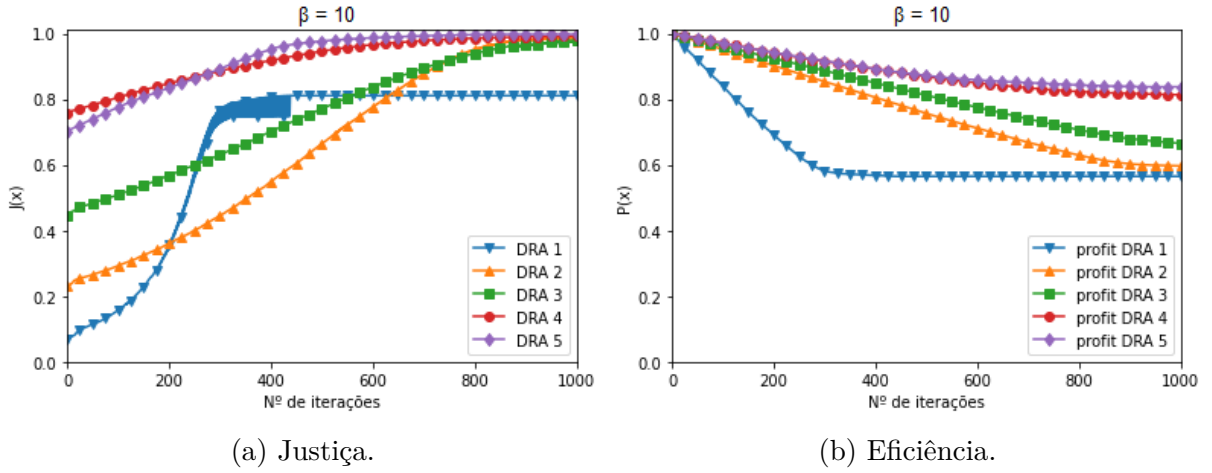
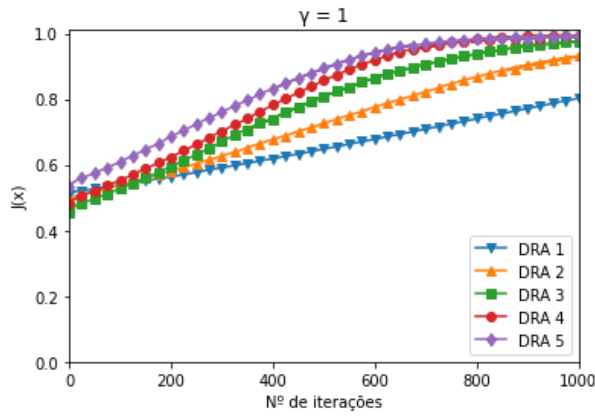


Figura 11 – Curvas de otimização de timeslots para $\beta = 10$ e $\gamma = 1$.

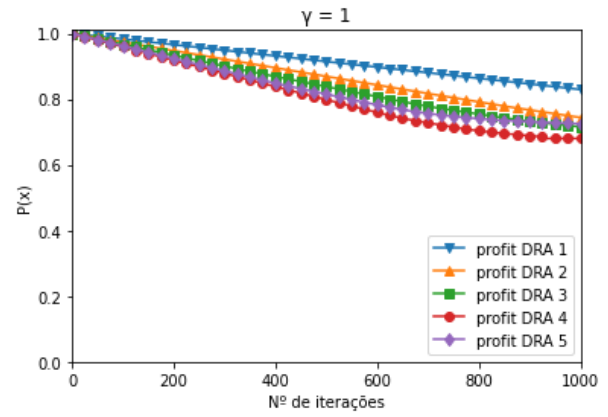
Considerando que os pontos de partida no eixo y representam as porcentagens iniciais de usuários completamente alocados por esquema DRA, observa-se pelos gráficos acima uma enorme discrepância na divisão de recursos para valores mais altos de Beta. A exemplo do item (a) da figura 11, nota-se que a DRA 1 consegue incorporar apenas 10% dos usuários ao final do algoritmo do mochileiro, sendo esses recursos já escassos posteriormente repartidos com os 90% restantes durante o algoritmo de justiça. Em comparação, observa-se um atendimento de 80% dos usuários da DRA 4 no mesmo instante de tempo, indicando uma maior quantidade de *timeslots* disponíveis e um maior nível de justiça entre os usuários do esquema.

4.2.2 Gamma (γ)

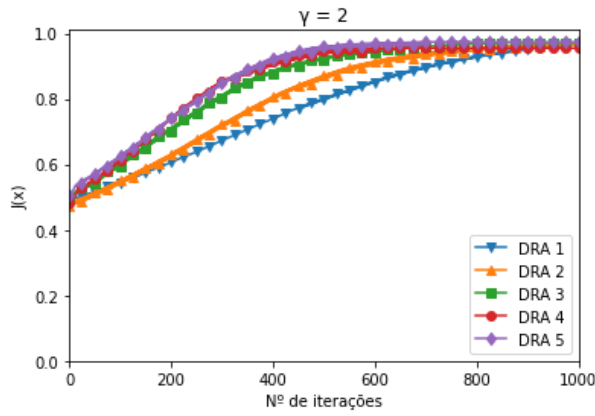
Semelhante ao Beta, Gamma governa a quantidade de *timeslots* transferida entre dois usuários de um mesmo esquema DRA. Valores altos de Gamma podem resultar em usuários recebendo mais *timeslots* que o solicitado, enquanto outros usuários permanecem com menos que o requisitado, gerando desperdício. Observa-se, no entanto, que a tendência de *timeslots* desperdiçados por esquema DRA tende a ser menor ou igual a Gamma. Há situações em que essa perda pode ser considerada irrelevante em face do ganho de velocidade de otimização.



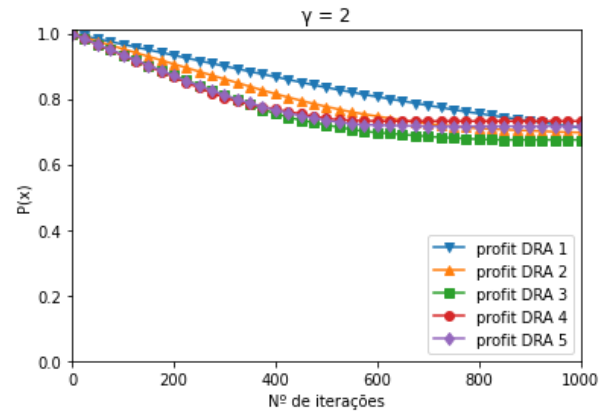
(a) Justiça.



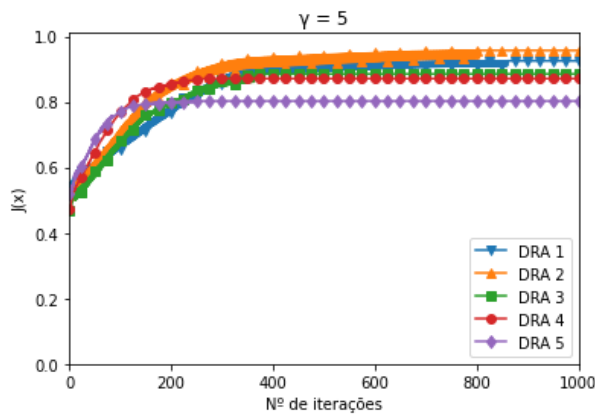
(b) Eficiência.

Figura 12 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 1$.

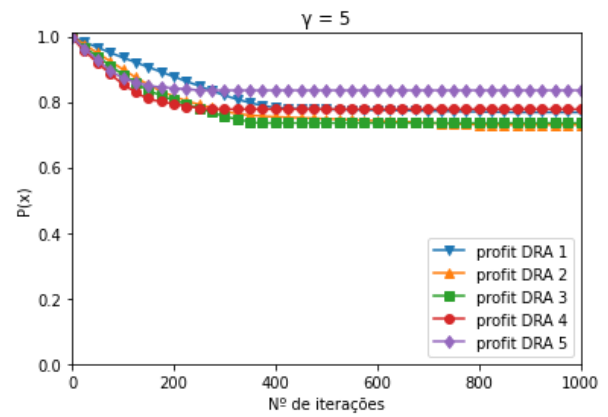
(a) Justiça.



(b) Eficiência.

Figura 13 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 2$.

(a) Justiça.



(b) Eficiência.

Figura 14 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 5$.

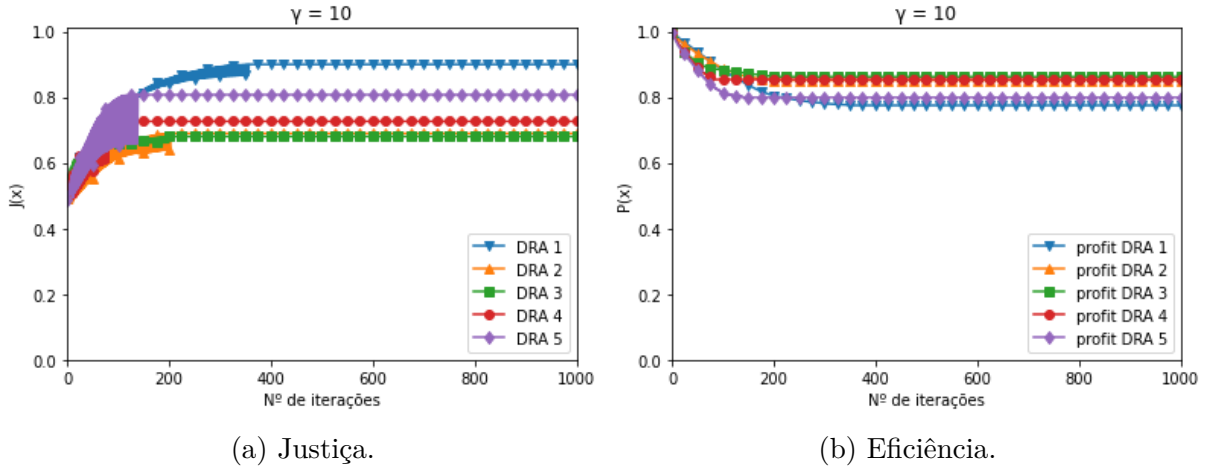


Figura 15 – Curvas de otimização de timeslots para $\beta = 1$ e $\gamma = 10$.

Conclui-se pelos gráficos acima que, quanto maior o Gamma, menor é o nível máximo de justiça alcançável pelo sistema e menor é a redução total de SNR média. As dispersões entre os níveis de justiça dos esquemas DRA podem ser explicadas pela quantidade total recursos presentes em cada esquema. A tendência é que, após o término da otimização de canais, o número de *timeslots* disponíveis seja maior para a DRA 1 e gradativamente decrescente para os esquemas seguintes, sendo a DRA 5 geralmente o esquema com a menor quantidade absoluta de *timeslots*. Em função da alta discretização durante a transferência de recursos vista em valores mais altos de Gamma, é natural que esquemas com menos *timeslots* sofram mais com o desperdício de recursos, como observado mais claramente nas figuras 14 e 15.

Descobrimos então que o nível de influência do parâmetro Gamma é diretamente proporcional à quantidade de *timeslots* disponíveis e inversamente proporcional à quantidade de usuários. Aproveitando essa nova relação observada, sugerimos a definição de um Gamma dinâmico capaz de ajustar uma taxa de transferência de *timeslots* específica para cada esquema DRA, maximizando as velocidades de aproximação de cada curva sem prejudicar os níveis máximo de justiça atingíveis. Dessa forma, propomos a equação 4.1 que define o valor ótimo de Gamma como a quantidade de *timeslots* disponíveis Q_{TS} dividida pela quantidade de terminais Q_t de um esquema DRA e multiplicada por uma variável de ajuste α responsável pelo controle da velocidade de aproximação do índice $J(x)$. Os resultados do uso do Gamma dinâmico podem ser observados nas figuras 16, 17, 18 e 19.

$$\gamma_d = \frac{Q_{TS}}{Q_t} \alpha, \alpha \in (0, 1) \quad (4.1)$$

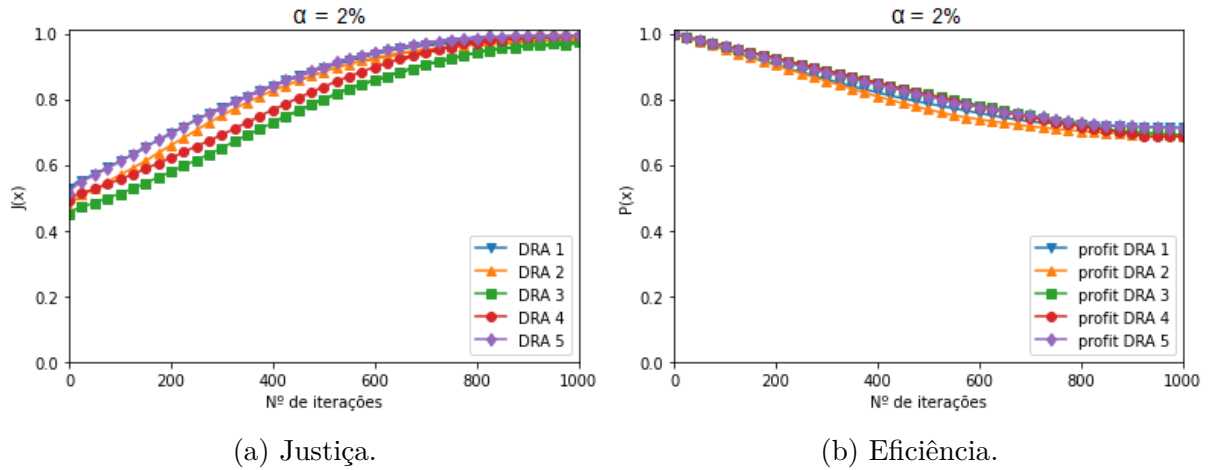


Figura 16 – Curvas de otimização de timeslots para $\gamma = 2\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.

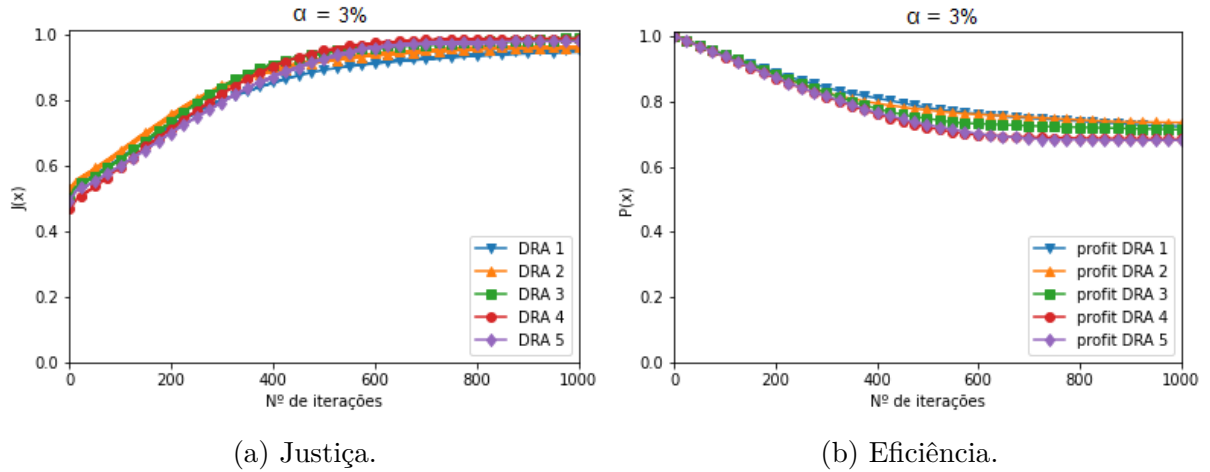


Figura 17 – Curvas de otimização de timeslots para $\gamma = 3\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.

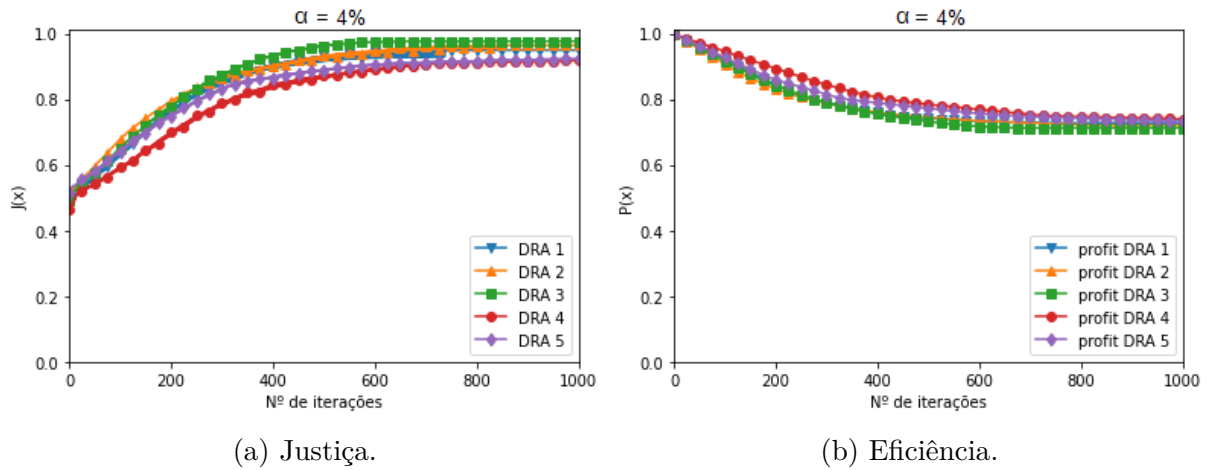


Figura 18 – Curvas de otimização de timeslots para $\gamma = 4\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.

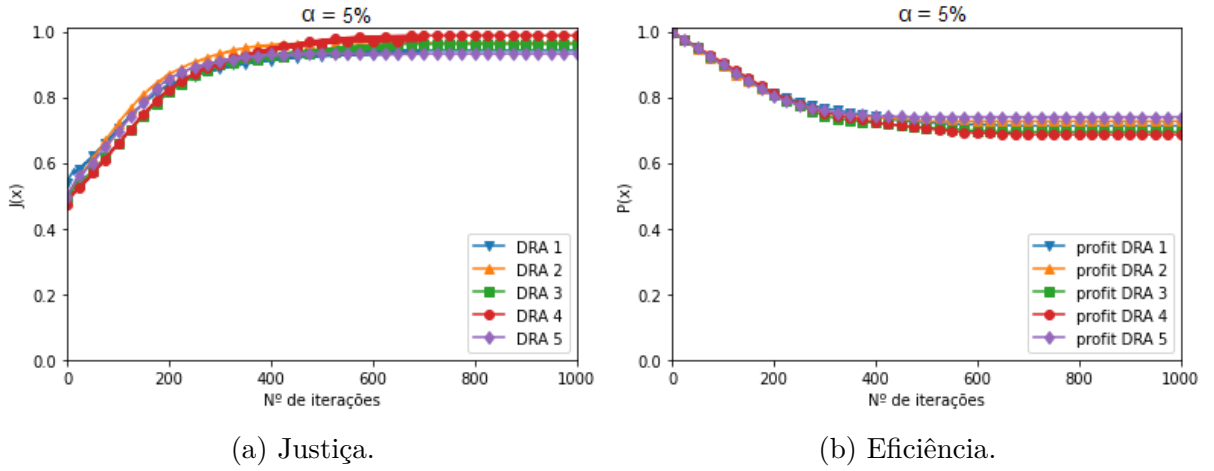


Figura 19 – Curvas de otimização de timeslots para $\gamma = 5\%$ da razão entre timeslots disponíveis e a quantidade de terminais de cada esquema DRA com $\beta = 1$.

Observa-se que o Gamma dinâmico permite que certos esquemas DRA, geralmente mais lentos e com maior quantidade total de *timeslots*, incrementem significativamente suas velocidades de convergência quando comparados aos testes com Gamma fixo, conseguindo acompanhar os esquemas mais rápidos no número de iterações necessárias para atingir o platô. Apesar de uma baixa diferença percentual entre as figuras 16 e 19, nota-se uma diferença considerável nas velocidades de aproximação sem efeitos perceptíveis na magnitude final de justiça.

4.2.3 Prioridade de usuários

Usuários prioritários fazem parte de um caso especial onde esses devem receber 100% dos recursos requisitados. Por este motivo, não há benefício em incluí-los no laço de otimização já que os mesmos não poderão transferir *timeslots* para outros usuários. A presença de um usuário prioritário significa que a quantidade de *timeslots* otimizáveis e, conseqüentemente, o nível máximo de justiça são reduzidos em comparação a um sistema sem graus de prioridade.

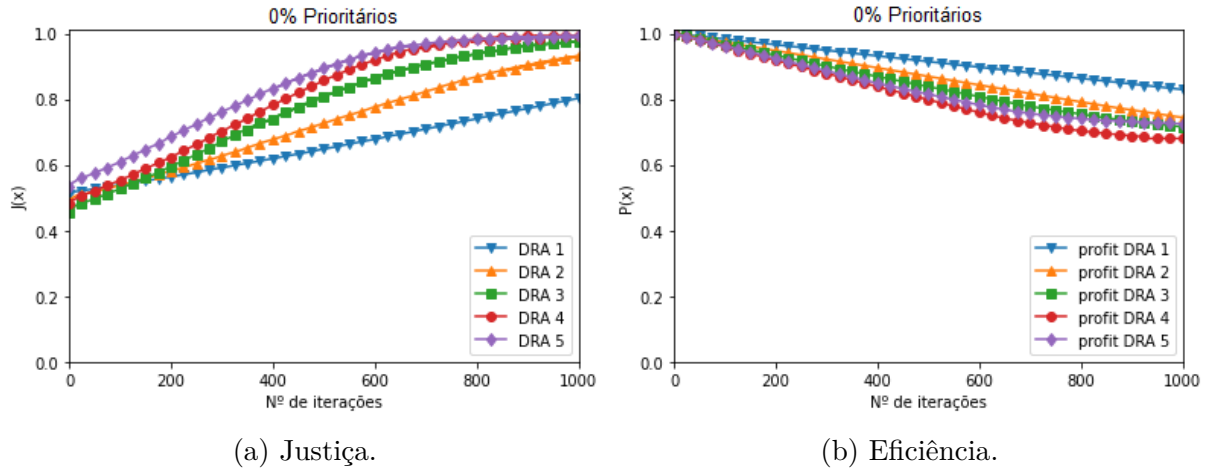


Figura 20 – Curvas de otimização de timeslots para 0% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.

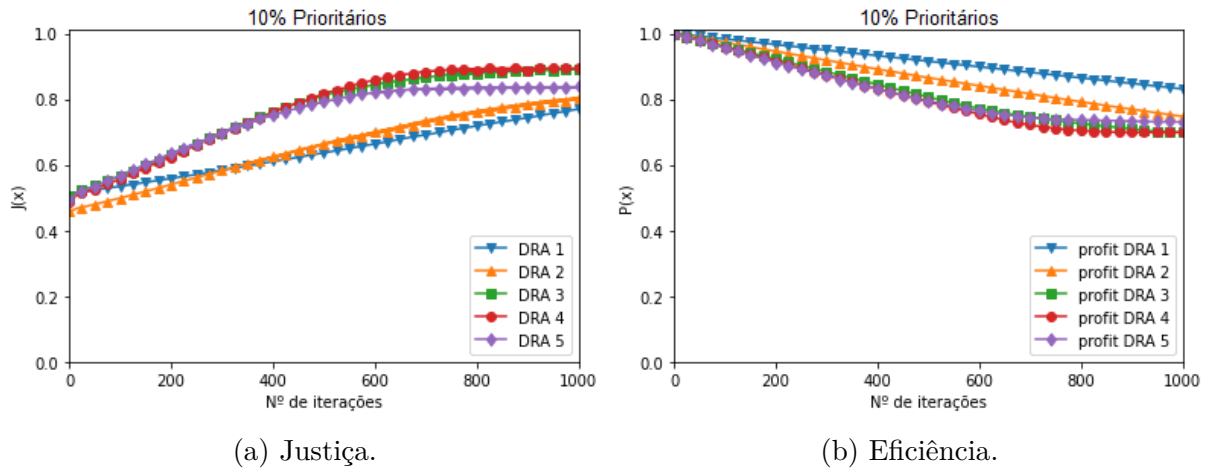


Figura 21 – Curvas de otimização de timeslots para 10% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.

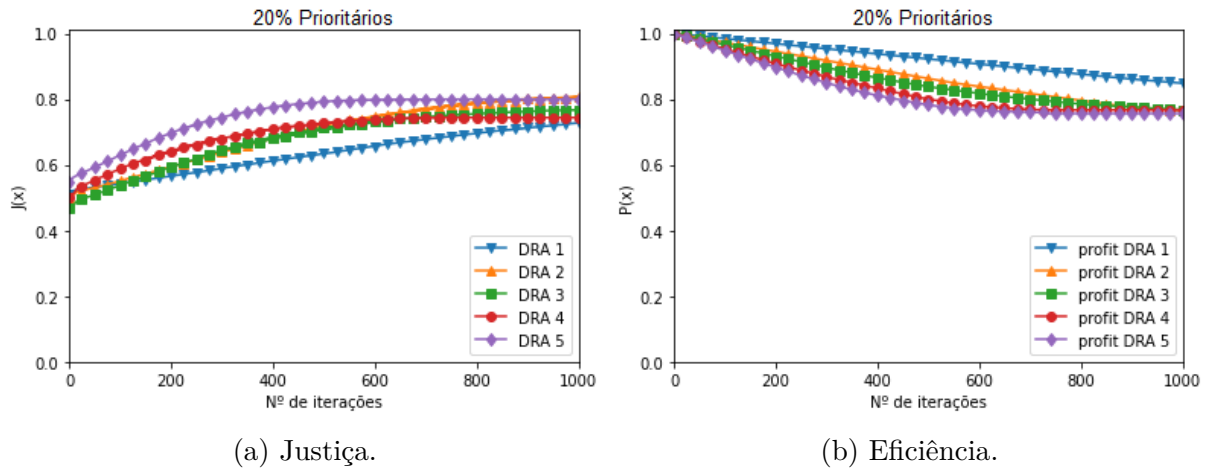


Figura 22 – Curvas de otimização de timeslots para 20% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.

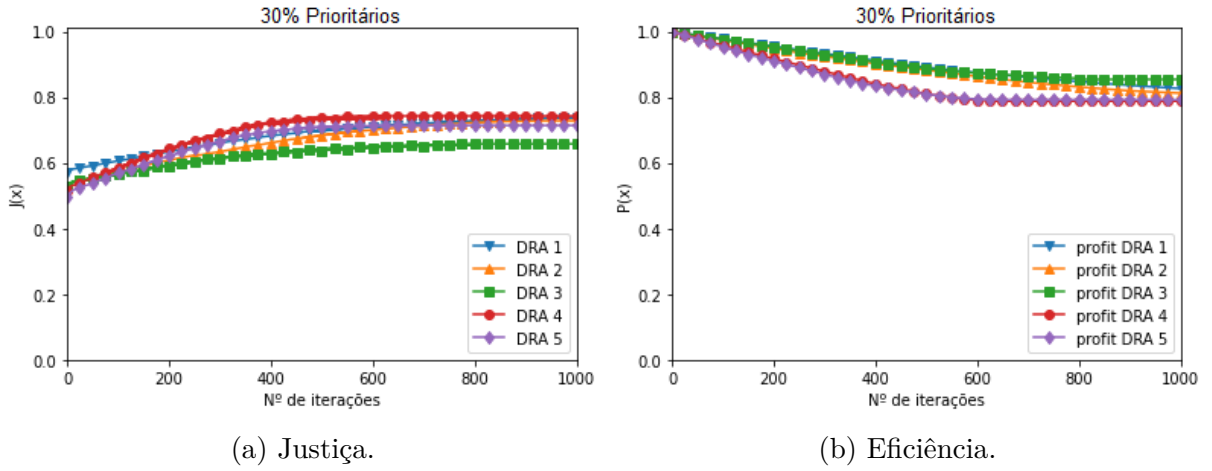


Figura 23 – Curvas de otimização de timeslots para 30% de usuários prioritários com $\beta = 1$ e $\gamma = 1$.

Como mencionado, usuários prioritários geram injustiças no sistema, confirmadas pelos gráficos acima, já que os mesmos não compartilham recursos com outros usuários. A definição da prioridade é dada antes dos algoritmos do mochileiro e de justiça e os *timeslots* reservados são removidos da capacidade livre. Apesar de não participarem ativamente da otimização, seus valores de SNR e justiça ainda são calculados e adicionados aos respectivos índices.

4.2.4 Payload e Quantidade de usuários

Excluindo-se os casos triviais onde o total de *timeslots* requisitados é menor que o número de *timeslots* disponíveis, sendo todos os usuários completamente alocados, temos dois extremos. No primeiro caso, analisa-se um sistema com poucos usuários e uma alta *payload* média. Para o segundo cenário, é estudado um sistema com excesso de usuários com baixa solicitação individual de *payload*.

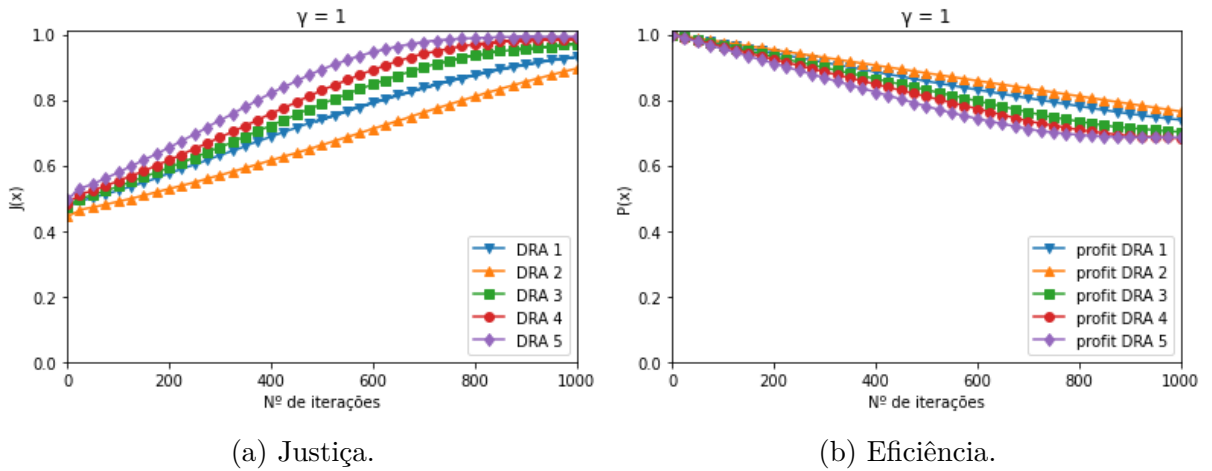


Figura 24 – Curvas de otimização de timeslots para sistemas com poucos usuários e alta *payload* média.

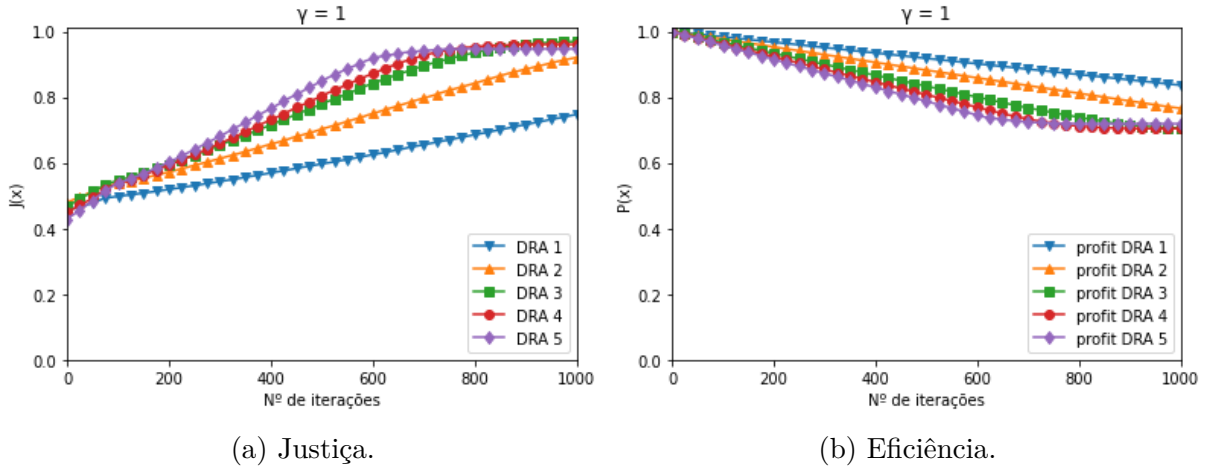


Figura 25 – Curvas de otimização de timeslots para sistemas com excesso de usuários e baixa payload média.

Por diferenças no tempo de execução do código, optou-se pela realização dos outros cenários de teste (Beta, Gamma e prioridade) sob o caso de poucos usuários com altas *payloads*. Muitas funções comparativas e multiplicativas utilizadas no algoritmo percorrem os vetores por completo, sendo os tamanhos destes muitas vezes relacionados à quantidade de usuários, mas nunca com o volume médio das *payloads*, tornando a otimização mais lenta.

Ambos os cenários foram calibrados para simular a mesma quantidade de recursos solicitados, sendo a relação entre a quantidade média de usuários e o volume médio de payload inversamente proporcionais. Para os gráficos da figura 24, a quantidade de usuários segue uma distribuição de Poisson para $\lambda = 50$, enquanto o volume de payload é definido por uma distribuição uniforme limitada em 45 kilobytes. Já na figura 25, utilizou-se $\lambda = 150$ para um limite de 15 kilobytes. Observou-se um aumento de 180% nos tempos de execução do primeiro para o segundo caso para um aumento de 200% na quantidade média de usuários.

A figura 26 analisa o desempenho dos seguintes cenários equivalentes: 1) 2 usuários solicitando 50 *timeslots* cada, com 50 *timeslots* disponibilizados pelo sistema e inicialmente alocados para o primeiro usuário; 2) 50 usuários solicitando 2 *timeslots* cada, com 50 *timeslots* disponíveis e inicialmente divididos entre os 25 primeiros usuários. Ambas as otimizações são concluídas ao total de 25 iterações sob a taxa de transferência de 1 *timeslot* por iteração ($\gamma = 1$).

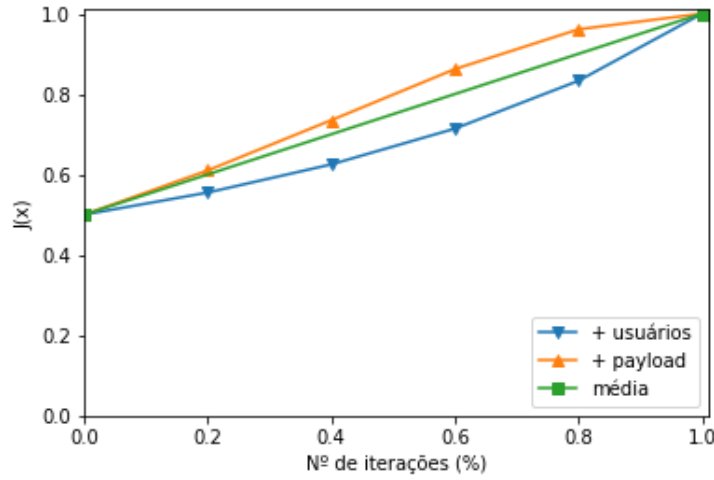


Figura 26 – Curvas de excesso de usuários vs excesso de payload para cenários de mesma magnitude.

Observa-se que cenários com poucos usuários e alta payload média (caso 1 e figura 24) começam com uma velocidade de otimização mais rápida que é desacelerada conforme o passar das iterações enquanto cenários com excesso de usuários com baixa payload média (caso 2 e figura 25) iniciam mais lentos e aceleram ao longo da otimização. Isso pode ser traçado à equação 3.1, onde a quantidade de usuários n é inversamente proporcional ao índice de justiça $J(x)$. Apesar das magnitudes de ambos os parâmetros não serem definidas pelo sistema, como Beta e Gamma, pode-se estabelecer limites para a quantidade de usuários e para o tamanho máximo de *payload* aceitos pelo sistema. Desta forma, o provedor pode administrar estes parâmetros externos a fim de controlar a velocidade de otimização e os tempos de execução do algoritmo.

4.3 Estudo Comparativo dos Algoritmos

Em questão de tempos de execução, a complexidade computacional do algoritmo BKP Greedy é de $O(L_k)$ para a alocação de recursos e $O(L_k \cdot \log L_k)$ para reposicionar os usuários por ordem de SNR decrescente totalizando $O\left[\sum_{k=1}^K L_k \cdot (1 + \log L_k)\right]$, onde K é o número de esquemas DRA e L_k é o número de usuários do esquema DRA k (MARTELLO; TOTH, 1990). Já para o algoritmo com critério de justiça é $O(N_F \cdot K + N_{TS} \cdot \sum_{k=1}^K L_k)$, onde o primeiro termo da soma retrata a otimização de *frames* para N_F iterações e o segundo termo é referente a otimização de *timeslots* com N_{TS} iterações (JUNG; SHIN; RYU, 2019).

Considerando apenas o caso de otimização de *timeslots*, comum para ambos os algoritmos, e uma quantidade de iterações igual a L_k para o algoritmo de justiça, podemos comparar os desempenhos de pior caso sob quantidades variáveis de usuários através das curvas de complexidade temporal na figura 27. Observa-se então que um número muito

maior de operações elementares são necessárias para a conclusão do algoritmo de justiça conforme é aumentado o número de usuários. Nota-se também que não há diferença significativa nos tempos de execução entre o uso exclusivo do algoritmo de justiça e o uso de ambos em conjunto, indicando um alto custo-benefício para utilização do algoritmo BKP em qualquer caso.

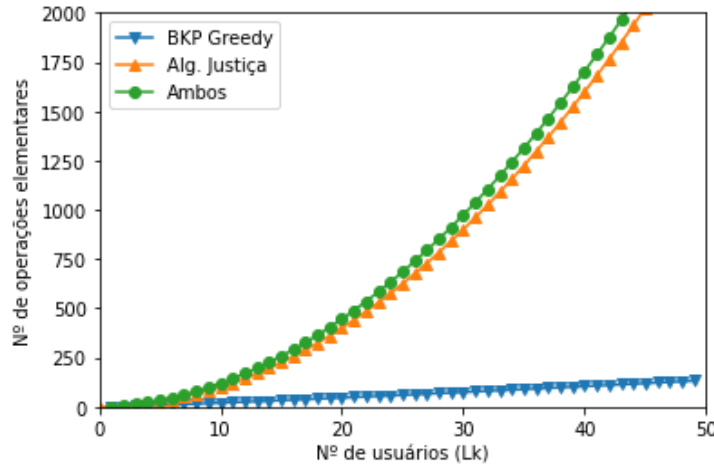


Figura 27 – Complexidade temporal relativa ao número de usuários.

Para casos triviais ou muito próximos da trivialidade, onde a demanda total de *timeslots* é igual a capacidade do sistema, o algoritmo do mochileiro é solução suficiente para alocação dos usuários, sendo consideravelmente mais rápido e capaz de retornar índices de justiça próximos de 100%. No entanto, o uso do algoritmo *Greedy* não foi considerado vantajoso para a otimização de *frames* pois não há uma definição contundente de lucro neste caso e a tendência do mesmo é fornecer uma solução que descarte certos esquemas em favor de outros, enfraquecendo o propósito da existência de múltiplas DRA's.

Os índices de justiça e SNR média demonstram ser afetados de maneira semelhante em magnitudes diferentes pelos parâmetros analisados. Observa-se também que a relação entre o ganho de justiça e a perda de SNR média não é proporcional nem determinística. Porém, o *trade-off* aparenta favorecer fortemente o índice de justiça na maioria dos casos já que a taxa de variação de SNR média é consistentemente mais lenta que a taxa de variação de justiça. A exemplo da figura 11, observa-se um aumento de 70% de justiça para a DRA 2 para uma queda de 40% na SNR média.

4.4 Comentários e Conclusões

A diferença entre os algoritmos estudados não se confina apenas aos resultados, pois ambos também servem a propósitos distintos. O algoritmo com critério de justiça favorece aos usuários pois seu objetivo é simplesmente permitir uma maior quantidade de acessos. Já o algoritmo do mochileiro é mais orientado ao provedor da rede pois garante

que os recursos sejam alocados para os terminais mais potentes, reduzindo o risco de perda de pacotes e, conseqüentemente, diminuindo a necessidade do reenvio de dados. Ao provedor ainda é dada a vantagem de selecionar qualquer combinação de limites inferiores e superiores de parada para os índices de SNR média e justiça, concluindo o processo de otimização em um número menor de iterações e atendendo aos requisitos mínimos esperados.

Vale mencionar que, para o algoritmo com critério de justiça, a existência de usuários que solicitam uma quantidade de *timeslots* igual ao valor de Γ é altamente perigosa para a otimização. Como explicitado anteriormente, *timeslots* são transferidos a uma taxa de γ sendo o doador, chamado detentor majoritário, o usuário com a maior razão '*timeslots* recebidos' por '*timeslots* solicitados' e o receptor, ou detentor minoritário, o extremo oposto com a menor proporção. Uma iteração do algoritmo seria suficiente para fazer um usuário que solicite γ *timeslots* receber 100% dos recursos solicitados ou perder todos os recursos que possui e ficar com 0%. Se dois ou mais usuários sob essa condição existirem no sistema, há alta probabilidade de surgir um *loop* na otimização onde ambos revezam os papéis de detentor majoritário e detentor minoritário, transferindo *timeslots* redundantemente entre si até o término do algoritmo, efetivamente empacando a otimização. Este problema pode ser evitado se provedor forçar o usuário a solicitar uma quantidade de *timeslot* diferente de γ , seja por um *timeslot* a mais ou a menos quando possível.

O uso do Γ dinâmico desenvolvido neste trabalho é uma solução eficiente para o algoritmo de justiça e permite o ajuste da taxa de transferência de *timeslots* entre usuários através de um parâmetro fixo. Por meio da definição de um α ótimo, pode-se reduzir consideravelmente o número de iterações necessárias para se atingir um nível desejável de justiça e torna o algoritmo flexível às quantidades variáveis de usuários e de canais disponíveis para cada esquema DRA a qualquer momento. Como observado nas figuras da seção 4.2.2, a velocidade é padronizada para todos os esquemas DRA e é independente das propriedades individuais de cada portadora. Com uma baixa quantidade de testes é possível estabelecer o valor ideal para α que habilite ao provedor sempre atingir um mesmo nível de *trade-off* entre velocidade de execução e qualidade do resultado final, tornando seu uso simples e vantajoso.

5 Conclusão e Trabalhos Futuros

5.1 Resultados Obtidos

As curvas geradas para os índices de justiça e SNR média apresentam resultados condizentes com o esperado, ambos percorrendo o gráfico em sentidos opostos e se estabilizando após o mesmo número de iterações. Parâmetros que afetam positivamente a velocidade de ganho de justiça também afetam positivamente a velocidade de perda de SNR média e a mesma análise vale para a concentração ou dispersão dos pontos ótimos entre os esquemas DRA.

Apesar das simulações neste relatório operarem sob uma janela de 1000 iterações, vale lembrar que a maioria dos testes foram executados com Gamma igual a 1 para esquemas DRA com milhares de *timeslots* disponíveis. O objetivo principal dessas análises era verificar o comportamento dos índices e o resultado final de alocação, mas valores muito baixos de Gamma somados a grandes quantidades de iterações são inviáveis para utilização em sistemas reais que requerem soluções de alta velocidade. Como visto nos testes com Gamma dinâmico, pode-se obter resultados excelentes com poucas centenas de iterações reafirmando a validade dos algoritmos.

Percebe-se ainda que nem sempre os níveis de justiça e SNR média se cruzam no gráfico como visto na figura 23, onde a DRA 3 atinge $J(x)$ máximo de 60% e $P(x)$ mínimo de 80%. Desta forma, ideias de alvejar um ponto de equilíbrio entre os dois índices são pouco praticáveis. Como mencionado anteriormente, o trade-off entre os índices favorece o algoritmo de justiça, com cada iteração do mesmo exercendo um impacto perceptivelmente maior no ganho de justiça do que na perda de SNR média. Por outro lado, o algoritmo do mochileiro possui um tempo de solução muito menor e é capaz de causar uma variação mais rápida em ambos os índices para uma quantidade menor de iterações.

5.2 Trabalhos Futuros

Os princípios de otimização aplicados neste relatório podem ser adaptados para qualquer sistema MF-TDMA ou semelhante. Cita-se sistemas híbridos de redes satelitais-terrestres que estão sendo estudados e desenvolvidos para o uso em 5G. Satélites multi-feixe poderão complementar e cooperar com torres de transmissão em solo em casos de sobrecarga e congestionamento da rede ou estender o alcance do serviço a zonas fora da cobertura das redes terrestres (ZANGAR; HENDAOU, 2018). Em razão das estimativas de um aumento de 10 a 100 vezes no número conexões por km^2 entre as tecnologias 4G

e 5G/6G respectivamente, espera-se um desenvolvimento mais amplo dos mecanismos de gestão de tráfego para as diversas modalidades de comunicação, sendo os conceitos base de justiça e ganância apresentados um ponto comum e reutilizável para qualquer caso.

Ainda dentro do contexto multifeixe, o feixe a ser utilizado pelo usuário pode ser considerado como uma dimensão adicional para o problema de alocação. Neste caso, cabe investigar se o algoritmo do mochileiro em sua versão N-dimensional mantém o bom desempenho com relação ao tempo computacional.

Planeja-se futuramente a continuação deste projeto com o objetivo de reduzir as otimizações de justiça e eficiência a uma única fórmula, diminuindo a complexidade computacional do algoritmo e oferecendo uma solução que trate ambos os parâmetros de maneira mais homogênea. Outro interesse é a flexibilização do parâmetro Gamma, com intenções de torná-lo variável durante as iterações do algoritmo de modo a agilizar ainda mais o procedimento.

Referências

- 3GPP TR 22.822 V16.0.0 (2018-06). *3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on using Satellite Access in 5G; Stage 1 (Release 16); Technical Report*. [S.l.], 2018. Citado na página 12.
- 3GPP TR 38.811 V15.0.0 (2018-06). *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on New Radio (NR) to support non terrestrial networks (Release 15); Technical Report*. [S.l.], 2016. Citado na página 12.
- CHEN, X. et al. Massive access for 5g and beyond. 02 2020. Citado na página 12.
- DUPE, J.-B. *Ordonnancement et gestion des ressources pour un système de télécommunications haut débit : Optimisation de la bande passante satellite*. Tese (Doutorado), 2015. Disponível em: <<http://www.theses.fr/2015INPT0136>>. Citado 2 vezes nas páginas 5 e 11.
- ETSI TS 101 545-1. *Digital video broadcasting (DVB); second generation DVB interactive satellite system (DVB-RCS2); Part 1: Overview and System Level Specification*. [S.l.], 2014. Citado 3 vezes nas páginas 5, 11 e 18.
- HOROWITZ; SAHNI. *Fundamentals of Computer Algorithms*. 2nd. ed. USA: Silicon Press, 1984. ISBN 0929306414. Citado na página 26.
- JAIN, R.; CHIU, D. M.; WR, H. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *CoRR*, cs.NI/9809099, 09 1984. Citado 3 vezes nas páginas 13, 19 e 20.
- JUNG, D.; SHIN, M.; RYU, J. Fairness-based superframe design and resource allocation for dynamic rate adaptation in dvb-rs2 satellite systems. *IEEE Communications Letters*, v. 23, n. 11, p. 2046–2049, Nov 2019. Citado 3 vezes nas páginas 14, 21 e 42.
- KARSU, ; MORTON, A. Inequity averse optimization in operational research. *European Journal of Operational Research*, v. 173, 02 2015. Citado na página 14.
- KELLERER, H.; PFERSCHY, U.; PISINGER, D. *Knapsack Problems*. [S.l.]: Springer, Berlin, Germany, 2004. Citado 5 vezes nas páginas 13, 19, 23, 25 e 29.
- LEE, K.-D. et al. Optimal scheduling for timeslot assignment in mftdma broadband satellite communications. In: . [S.l.: s.n.], 2002. v. 3, p. 1560 – 1564 vol.3. ISBN 0-7803-7467-3. Citado 3 vezes nas páginas 5, 13 e 17.
- LIOLIS, K. et al. Use cases and scenarios of 5g integrated satellite-terrestrial networks for enhanced mobile broadband: The sat5g approach. *International Journal of Satellite Communications and Networking*, v. 37, n. 2, p. 91–112, 2019. Citado na página 12.
- MARTELLO, S.; TOTH, P. *Knapsack Problems: Algorithms and Computer Implementations*. USA: John Wiley and Sons, Inc., 1990. ISBN 0471924202. Citado 2 vezes nas páginas 13 e 42.

- MELONI, A.; MURRONI, M. Random access in dvb-rcs2: Design and dynamic control for congestion avoidance. *Broadcasting, IEEE Transactions on*, v. 60, p. 16–28, 03 2014. Citado na página 17.
- PIETRABISSA, A.; FIASCHETTI, A. Dynamic uplink frame optimization with acm in dvb-rcs2 satellite networks. In: *2012 IEEE First AESS European Conference on Satellite Telecommunications (ESTEL)*. [S.l.: s.n.], 2012. p. 1–7. Citado 5 vezes nas páginas 5, 11, 12, 14 e 16.
- RAWLS, J. *A Theory of Justice*. 1. ed. Cambridge, Massachussets: Belknap Press of Harvard University Press, 1971. ISBN 0-674-88014-5. Citado na página 19.
- RENDON-MORALES, E. et al. Adaptive packet scheduling for the support of qos over dvb-s2 satellite systems. In: . [S.l.: s.n.], 2011. v. 6649, p. 15–26. ISBN 978-3-642-21559-9. Citado na página 20.
- WANG, X. et al. A tradeoff resource allocation based on mf-tdma scheme in the multibeam data relay satellite systems. *International Journal of Satellite Communications and Networking*, v. 37, 09 2018. Citado na página 14.
- YAZDANI, N. Multi-frequency time-division multiple-access (mf-tdma) resource packing. In: *MILCOM 2008 - 2008 IEEE Military Communications Conference*. [S.l.: s.n.], 2008. p. 1–8. Citado na página 13.
- ZANGAR, N.; HENDAOU, S. Leveraging multiuser diversity for adaptive hybrid satellite-lte downlink scheduler (h-mudos) in emerging 5g-satellite network. *International Journal of Satellite Communications and Networking*, v. 37, 08 2018. Citado 2 vezes nas páginas 15 e 45.